Fitting a Quadratic Surface to Three Dimensional Data

Bennett Groshong
Griff Bilbro
Wesley Snyder

Center for Communications and Signal Processing
Electrical and Computer Engineering Department
North Carolina State University

October 1989

CCSP TR-89/17
1. Introduction

There is a need in 3-d image processing and analysis to be able to represent regions in an image with parametric equations. We need a representation of the data that is concise, allowing parts of the image to be represented compactly as we move towards a global description. The representation used should be resistant to noise in the image. On the other hand, it should be rich enough to describe well the set of surfaces/objects in which we are interested. In this paper, we will describe the rationale and the methodology for fitting an analytic surface to three dimensional data points. In the proceedings of the 1983 IEEE Conference on Computer Vision and Pattern Recognition, Faugeras\(^1\) alluded to a method for fitting quadric surfaces which appears similar to the method described in this paper. In order for the results of this work to be of general use to the research community, we have attempted to both justify and thoroughly describe the methods contained therein.

2. Desiridata

We desire an accurate analytic description of the surface for segmentation purposes so that we can take the first and second partial derivatives and obtain meaningful measures of such features as range discontinuities, surface normals and surface curvature. This same function applied to larger regions in the image is useful for surface and object description, model matching, and pose determination.

One surface representation spanning the trade-offs between noise immunity, compactness, and descriptive richness is the general quadric. A general quadric is defined\(^5\) as the locus of points satisfying

\[
F(x,y,z) = ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0
\]

Limiting the equation to second degree provides good noise performance. The quadric form that \(F(x,y,z)\) represents is invariant to rotation and translation in all three dimensions. The coefficients change in well known ways in different coordinate systems. The quadric equation is fairly compact, requiring just ten parameters. Yet it is rich in
descriptive power, covering a number of surfaces of interest in industrial applications. For instance, intersecting planes, spheres, cylinders, cones, and ellipsoids can all be represented exactly with quadrics.

3. Problem Description

We wish to solve the following problem: Given a set of 3-d data points $S = (x_i, y_i, z_i)$, determine the ten coefficients that define the analytic quadric surface (eqn. 1) which best fits them. We make the following assumptions about the input data set $S$. First we make the assumption that the data is a set of range image samples drawn from a single surface which can be described by a quadric equation. In other words, the set $S$ comes from part of a single quadric surface. This assumption includes the sensor noise and quantization artefacts. The solution we will formulate for quadric surfaces does not include planar surfaces. Planar surfaces are common in industrial scenes, and will be dealt with later so that the complete package solves for both planar and quadric surfaces.

4. Minimization of Error

We wish to produce the best description of the data possible with the quadric equation (1). We shall define the best description to be that which minimizes the mean squared error (MSE) between the data and the quadric over the set $S$. Let us write (1) down in vector notation.

$$F(x,y,z) = a^T p = 0$$

where

$$a^T = \begin{bmatrix} a & b & c & 2f & 2g & 2h & 2p & 2q & 2r & d \end{bmatrix}$$

and

$$p^T = \begin{bmatrix} x^2 & y^2 & z^2 & yz & zx & xy & x & y & z & 1 \end{bmatrix}$$

We formulate the minimization of mean squared error by first noting that eqn. (1) has the form $F(x,y,z) = 0$, rather than the more familiar $z = F(x,y)$. This observation allows us to write the minimization as
\[ E = \min_a \sum_S ||F||_2 \]  

In vector notation, this can be written

\[ E = \min_a \sum_S a^T p p^T a = \min_a a^T R a \]  

Where \( R \) is the scatter matrix for the data set equal to

\[ R = \sum_S p p^T \]  

To minimize, we take the partial derivatives of the error, which we have labeled \( E \), with respect to the coefficient vector \( a \) and set them equal to zero (Nobel & Daniel's section 2.6).

\[ \frac{\partial E}{\partial a} = 2R a = 0 \]  

Unfortunately, this produces the trivial solution of \( a = 0 \). Since this solution is useless, we must find a way to constrain the solution for \( a \) to be non zero, yet still be a quadric form.

5. Constraint Formulation

We are finding the minimum of \( a^T R a \) with respect to \( a \) subject to some constraint. We can write this as "Minimize the function \( G(a) \) subject to the constraint \( K(a) = k \), where

\[ G(a) = a^T R a \]  

\[ K(a) = a^T K a \]  

where \( K \) is an as yet undetermined constraint matrix for the solution on \( a \). To attack this problem using Lagrange's method (Taylor & Mann pp 182-183) we write the function

\[ u = G(a) - \lambda K(a) \]  

where \( \lambda \) is an undetermined constant. We want to find the minimum solution for \( u \), so we form

\[ \frac{\partial u}{\partial a} = 2(R - \lambda K)a = 0 \]  

now we solve
\[
\frac{\partial u}{\partial a} = 0 \quad (13)
\]

and

\[
K(a) = k \quad (14)
\]
simultaneously to find \( a \) and \( \lambda \) giving us a minimum solution.

6. Constraint Derivation

We wish to formulate the constraint \( K(a) \) such that it gives us a non-zero solution for \( a \) for all of the quadric surface forms we are interested in. More importantly, our solution \( a^T p \) should describe the same quadric surface under the coordinate transformations of rotation and translation.

Let us write the quadric equation in a more geometrically intuitive form to determine what function of the coefficient vector \( a \) might be invariant under translation and rotation.

\[
F(x,y,z) = F(v) = v^T D v + 2v^T q + d = 0 \quad (15)
\]
where

\[
v = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (16)
\]

\[
D = \begin{bmatrix}
    a & h & g \\
    h & b & f \\
    g & f & c
\end{bmatrix} \quad (17)
\]

\[
q = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (18)
\]

First let us consider translation. If we perform an arbitrary translation on \( v \) we write

\[
v' = v + \alpha \quad (19)
\]

we get

\[
F(v') = v'^T D v' + 2v'^T q' + d' \quad (20)
\]

where

\[
q' = q + 2D \alpha \quad (21)
\]
\[ d' = d + \alpha^T D \alpha + 2\alpha^T q \]

and we see that the second order terms matrix D is invariant under translation. Both \( q \) and \( d \) change, so the derived constraint should not include them.

Now let us consider rotation. If we perform an arbitrary rotation upon our coordinate system \( v \) the new coordinate system \( v' \) is given by \( v' = Ov \) where \( O \) is an orthogonal rotation matrix.\(^3\) now

\[ F(v') = v'^T O^T D O v' + v'^T O^T q + d = 0 \quad (23) \]

\[ F(v') = v'^T D' v' + q' + d = 0 \quad (24) \]

So to form a constraint that is invariant to rotation and translation, we need to find out what is invariant about \( D \).

Since \( D \) is a real, symmetric matrix, we can rotate \( D \) so that the result \( D' \) is a diagonal matrix with the eigenvalues of \( D \) on the diagonal. In fact, these eigenvalues are the only invariants of \( D \) under rotation.

We want some function of the eigenvalues of \( D \), \( f(\lambda) \), that will allow all of the quadrics we are interested in. From Jeger and Eckmann\(^2\) (pp 186 - 199), we can see that the various real quadric surfaces result in eigenvalues that may be positive, negative, or zero, but not all simultaneously zero. We also want the constraint to be quadratic in form so that when we substitute into the minimizing equation (12), we will get a linear equation in \( a \) that we can solve for \( a \). A good choice\(^1\) for our invariant constraint \( f(\lambda) \) is

\[ f(\lambda) = \sum_i \lambda_i^2 = 1 \quad (25) \]

We can write this in terms of \( D \) (eqn 17) as

\[ \sum_i \lambda_i^2 = \text{tr}(D^2) = a^2 + b^2 + c^2 + 2f^2 + 2g^2 + 2h^2 \quad (26) \]

Now let us write this in the form of eqn. 10:

\[ \text{tr} \left( D^2 \right) = a^T \begin{bmatrix} K_2 & 0 \\ 0 & 0 \end{bmatrix} a \quad (27) \]

Where \( a \) is defined in (eqn. 3) and where the constraint matrix \( K_2 \) is
7. Mathematics and Computational Considerations

We are solving (12) for \( a \) giving minimum error given the constraint \( K \). Rewritten, this is

\[
Ra = \lambda K a
\]  
(28)

Since part of our constraint matrix is zero, we will divide the problem into submatrices.

\[
\begin{bmatrix}
C & B \\
B^T & A
\end{bmatrix}
\begin{bmatrix}
\beta \\
\alpha
\end{bmatrix}
= \lambda
\begin{bmatrix}
K_2 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\beta \\
\alpha
\end{bmatrix}
\]  
(29)

where \( C \) is the \( 6 \times 6 \) scatter matrix for the quadratic terms, \( B \) is the \( 6 \times 4 \) scatter matrix of the mixed terms, and \( A \) is the \( 4 \times 4 \) scatter matrix of the linear and constant terms. \( \beta \) is the \( 6 \times 1 \) vector of the quadratic coefficients, \( \alpha \) is the \( 4 \times 1 \) vector of the linear and constant coefficients. \( K_2 \) is the \( 6 \times 6 \) constraint matrix with the non zero terms of \( K \) on the diagonal. The lower right 0 in \( K \) indicates that \( \alpha \) is not constrained.

Now let us write this as two equations.

\[
C \beta + B \alpha = \lambda K_2 \beta
\]  
(30)

\[
B^T \beta + A \alpha = 0
\]  
(31)

Solving (31) for \( \alpha \)

\[
\alpha = -A^{-1} B^T \beta
\]  
(32)

We know \( A^{-1} \) exists, since \( A \) is a real, symmetric matrix and the data is not planar, linear, or a point. Now substitute into (eqn. 30)
\[ C\beta - BA^{-1}B^T\beta = \lambda K_2\beta \]  \hspace{1cm} (33)

Simplifying a bit, we have
\[ (C - BA^{-1}B^T)\beta = \lambda K_2\beta \]  \hspace{1cm} (34)

Labeling \((C - BA^{-1}B^T)\) as \(M\), we have
\[ M\beta = \lambda K_2\beta \]  \hspace{1cm} (35)

This appears to be similar to an eigenvalue problem. We need to reformulate it slightly to put it in correct form for an eigenvalue solution. If we write \(K_2\) as \(H^2\) where

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \frac{1}{\sqrt{2}}
\end{bmatrix}
\]  \hspace{1cm} (36)

and

\[
H^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \sqrt{2}
\end{bmatrix}
\]  \hspace{1cm} (37)

we can write eqn. 35 as
\[ M\beta = \lambda HH\beta \]  \hspace{1cm} (38)

Operating on (eqn. 38)
\[ H^{-1}MH^{-1}H\beta = \lambda H\beta \]  \hspace{1cm} (39)

Now if we define \(\beta' = H\beta\) and \(M' = H^{-1}MH^{-1}\) where \(M'\) is a real symmetric matrix, We can write (39) as
\[ M'\beta' = \lambda \beta' \] (40)

The problem is now in the form amenable to eigenvalue solution. We find the eigenvalues and vectors of \( M' \) where there are six \( \lambda_i \)'s and six corresponding \( \beta_i' \)'s. For the minimum error solution, we choose the eigenvector corresponding to the smallest eigenvalue.

Now we convert back to unprime space, writing

\[ \beta_i = H^{-1}\beta_i' \] (41)

and solve for

\[ \alpha_i = -A^{-1}B^T\beta_i \] (42)

and we have our solution, the coefficient vector giving the minimum error fit to the data

\[ a_i = [\beta_i \mid \alpha_i] \] (43)

8. Fitting Planes

Earlier in section 3, we mentioned the need to fit planes. The constrained solution we have developed does not guarantee non zero solutions for planes. We will have to solve a separate but similar problem to find the best fitting plane given the data. Following the same process of development as for the quadric solution, we write the equation for the plane as

\[ F(x,y,z) = px + qy + rz + d = F(\mathbf{v}) = \mathbf{v}^T\mathbf{q} + d = 0 \] (44)

Again we must avoid the trivial zero solution by using a constraint. We choose

\[ K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \] (45)

as our constraint and write an equation similar to eqn.(28) for the plane solution.

\[ R\beta + \lambda K\beta = 0 \] (46)

Breaking the problem into submatrices, we have an equation of the form
(eqn. 29) where the A, B, and C matrices contain the linear and constant scatter terms. A is $1 \times 1$, B is $3 \times 1$, and C is $3 \times 3$. $K_2$ is a $3 \times 3$ identity matrix. Now we solve

$$M\beta = \lambda \beta$$

(47)

In a similar manner to section 7, we solve for the eigenvector corresponding to the smallest eigenvalue, back substitute to find $\alpha$, and we have our coefficients for the plane.

9. Measurement of Error

The smallest eigenvalue in each of the plane and quadric cases represents the total squared error in the fit. This squared error can be written in matrix form as $G(a) = a^T R a$ (eqn. 9) If we break $R$ down into submatrices as in equation 29, the squared error becomes

$$G(a) = \beta^T C \beta - 2\beta^T B \alpha + \alpha^T A \alpha$$

(48)

The set of six eigenvectors $(\beta_i)$ found for equation 40 form a basis for any $\beta$. We can write any $\beta$ as a sum of the scaled eigenvectors:

$$\beta = \sum_{i=1}^{6} \gamma_i \beta_i$$

(49)

where the $\gamma_i$'s represent the scale factors. We note that $\alpha$ is simply a function of $\beta$ (equation 32). Writing $\alpha$ as a function of the eigenvectors, we have

$$\alpha = -\sum_{i=1}^{6} \gamma_i A^{-1} B^T \beta_i$$

(50)

Expanding equation (48) by substituting equations (49) and (50), we have

$$G(a) = \sum_{i=1}^{6} \gamma_i \beta_i^T C \sum_{j=1}^{6} \gamma_j \beta_j$$

$$- 2 \sum_{i=1}^{6} \gamma_i \beta_i^T B \sum_{j=1}^{6} \gamma_j A^{-1} B^T \beta_j$$

$$+ \sum_{i=1}^{6} \gamma_i \beta_i^T B A^{-1} A \sum_{j=1}^{6} \gamma_j A^{-1} B^T \beta_j$$

(51)

Shifting the scalars and summations in (51) to the left,
\[ G(a) = \sum_{i=1}^{6} \sum_{j=1}^{6} \gamma_i \gamma_j \beta_i^T C \beta_j \] (52)

\[- 2 \sum_{i=1}^{6} \sum_{j=1}^{6} \gamma_i \gamma_j \beta_i^T B A^{-1} B^T \beta_j \]

\[ + \sum_{i=1}^{6} \sum_{j=1}^{6} \gamma_i \gamma_j \beta_i^T B A^{-1} A \beta_j \]

Now we group the constant matrices in (52)

\[ G(a) = \sum_{i=1}^{6} \sum_{j=1}^{6} \gamma_i \gamma_j \beta_i^T \left[ C - BA^{-1} B^T \right] \beta_j \] (53)

To simplify this equation, we note that we have the same form as equation 34. making the substitution simplifies (53) further.

\[ G(a) = \sum_{i=1}^{6} \sum_{j=1}^{6} \gamma_i \gamma_i \lambda_j \beta_i^T K_2 \beta_j \] (54)

Now we go back to the eigensystem solution \( M' \beta' = \lambda \beta' \) (eqn 40). The solution of this equation is subject to the classical constraints that

\[ \beta'_i^T \beta'_j = 1, \ i = j \quad \beta'_i \beta'_j = 0, \ i \neq j \] (55)

Noting that

\[ \beta'_i^T \beta'_i = \beta_i^T K_2 \beta_i \] (56)

(from eqn. 41) we further simplify eqn 54.

\[ G(a) = \sum_{i=1}^{6} \gamma_i \beta_i^2 \lambda_i \] (57)

If we choose \( \beta = \beta_i \) corresponding to the smallest eigenvalue, \( G(a) = \lambda_{\text{small}} \). We have shown that \( \lambda_{\text{small}} \) represents the total squared error in the quadric fit to the data.

A measure of fit error that is more meaningful in geometric terms is

\[ \text{err} = \left( \frac{\lambda_{\text{smallest}}}{N_{\text{points}}} \right)^{\frac{1}{2}} \] (58)

This gives us the rms error per point, which is not sensitive to the number of points. It is not, however phrased in terms of surface area. The use of rms error per unit surface area might be a better error measure, but has not as yet been tested.
10. Algorithm

To perform a fit to quadric or planar surfaces, we write the following algorithm.

10.1. Scatter Matrix Formation

Form the $10 \times 10$ scatter matrix $R$ summed over the data (eqn. 7).

```
ZERO(SCATTER_MATRIX)
WHILE(data)
    FORM_P_VECTOR(P, (x_i, y_i, z_i))
    FOR(J = 1 TO 10){
        FOR(K = 1 TO 10){
            SCATTER_MATRIX(I,J) += P(I,J)^T P(I,J)
        }
    }
```

10.2. Minimum MSE Fit to a Plane

Form the matrices $A_{1 \times 1}, B_{4 \times 1}, C_{4 \times 4}$ by dividing the scatter matrix up as in equation 29.

```
SPLIT(SCATTER_MATRIX, A, B, C)
```

Manipulate the matrices to get the problem into a form for eigensystem solution (eqns. 45 - 47) and solve the eigensystem problem.

```
INVERSE(A, AI)
TRANSPOSE(B, BT)
MULTMAT(B, AI, TEMP1)
MULTMAT(TEMP1, BT, TEMP2)
ADDMAT(TEMP2, C, TEMP2)
EIGEN(TEMP2, LAMBDA, EIGEN_VECTORS)
```

If the fit error is small enough, return a plane fit and its associated error.

```
IF(LAMBDA_SMALL(LAMBDA) < T){
    RETURN( CALC_PLANE_EQUATION( PLANE ) )
}
```
Else, compute the quadric fit coefficients, returning them and the error measure.

10.3. Minimum MSE Fit to a Quadric

The solution is of the same general form as for the plane. Form the matrices $A_{4 \times 4}, B_{6 \times 4}, C_{6 \times 6}$ by dividing the scatter matrix up as in equation 29.

\[
\text{SPLIT( SCATTER\_MATRIX, A, B, C )}
\]

Manipulate the matrices to get the problem into a form for eigensystem solution (eqns. 34 - 40) and solve the eigensystem problem.

\[
\text{INVERSE( A, AI )}
\]
\[
\text{TRANSPOSE( B, BT )}
\]
\[
\text{MULTMAT( B, AI, TEMP1 )}
\]
\[
\text{MULTMAT( TEMP1, BT, TEMP2 )}
\]
\[
\text{ADDMAT( TEMP2, C, TEMP2 )}
\]
\[
\text{MULTMAT( H\_CONSTRAINT\_INV, TEMP2, TEMP2 )}
\]
\[
\text{MULTMAT( TEMP2, H\_CONSTRAINT\_INV, TEMP2 )}
\]
\[
\text{EIGEN( TEMP2, LAMBDA, EIGEN\_VECTORS )}
\]
\[
\text{RETURN( CALC\_QUADRIC\_EQUATION( LAMBDA, EIGEN\_VECTORS ) )}
\]

11. Conclusion

This algorithm has been coded and gives excellent results for the surfaces tried. It is computationally intensive, requiring approximately $N \times 131 + 1000$ multiplies for a quadric fit where $N$ is the number of data points. Performance of the plane fit - quadric fit discriminator under adverse conditions is questionable. Consider the very short section of a cylinder illustrated in figure 1. It may be more useful to return both the plane and quadric fit parameters, letting higher level processes determine the model used for the surface. For an example fit to a parabolic cylinder, see figure 1. The plot is of a one dimensional slice through

12
the cylinder perpendicular to its axis.

Figure 1. Quadric Fit to a Parabolic Cylinder