Relative Orientation Revisited

Berthold K.P. Horn

15 March 1990

Abstract: *Relative Orientation is the recovery of the position and orientation of one imaging system relative to another from correspondences between five or more ray pairs. It is one of four core problems in photogrammetry and is of central importance in binocular stereo, as well as in long range motion vision. While five ray correspondences are sufficient to yield a finite number of solutions, more than five correspondences are used in practice to ensure an accurate solution using least squares methods. Most iterative schemes for minimizing the sum of squares of weighted errors require a good guess as a starting value. The author has previously published a method that finds the best solution without requiring an initial guess. In this paper an even simpler method is presented that utilizes the representation of rotations by unit quaternions.*

1. Introduction

Relative orientation is one of the central problems in photogrammetry and has attracted attention for more than a hundred years [Hauck 1883] [Finsterwalder 1899]. We briefly review the problem here. For additional background material and list of references see [Horn 90].

The coordinates of corresponding points in two images can be used to determine the positions of points in the environment, provided that the position and orientation of one of the cameras with respect to the other is known. Given the internal geometry of the cameras, including its principal distance and the location of the principal point, rays can be constructed by connecting the points in the images to their corresponding projection centers. These rays, when extended, intersect at the point in the scene that gave rise to the image points. This is how binocular stereo data is used to determine the positions of points in the environment after the correspondence problem has been solved.

In both binocular stereo and large displacement motion vision analysis, it is necessary to first determine the *relative orientation* of one camera with respect to the other. The relative orientation can be found if a sufficiently large set of pairs of corresponding rays have been identified.

Let us use the terms *left* and *right* to identify the two cameras (in the case of the application to long range motion vision these will be the camera positions and orientations corresponding to the earlier and the later frames respectively). The ray from the center of projection of the left camera to the center of projection of the right camera is called the *baseline*. A coordinate system can be erected at each projection center, with one axis along the optical axis, that is, perpendicular to the image plane. The other two axes are in each case parallel to two convenient orthogonal directions in the image plane (such as the edges of the image, or lines connecting pairs of fiducial marks). The rotation of the left camera coordinate system with respect to the right is called the *orientation*.

Note that we cannot determine the length of the baseline without knowledge about the length of a line in the scene, since the ray directions are unchanged if we scale all of the distances in the scene and the baseline by the same positive scale-factor. This means that we should treat the baseline as a unit vector, and that there are really only five unknowns—three for the rotation and two for the direction of the baseline[1].

It has long been known that five sets of ray pairs are required to obtain a finite number of solutions of the relative orientation problem [Finsterwalder 1899]. In practice one measures more than five pairs of rays so that least squares methods can be used to obtain more accurate results. Several iterative schemes are in use to find solutions (see text books on stereophotogrammetry as well as references in [Horn 90]). Most require a good initial guess, and some do not work well unless the surface being viewed is approximately planar and perpendicular to the viewing direction.

The author has previously given an iterative scheme for solving the least-squares relative-orientation problem that does not require a good initial guess, and that works well even when the surface is not approximately planar [Horn 87b, 90]. Here a new formulation of the coplanarity condition is given using unit quaternion notation to represent rotation. A new iterative scheme based on this representation has been implemented and found to be both reliable and faster than the previous method. The new formulation of the coplanarity condition also suggests better continuation methods for solving the special case when there are only five ray pairs, and leads to a short proof that there can be at most twenty solutions in this case.

1.1 New Expression for the Coplanarity Condition

The volume of the parallelipiped formed by three vectors is equal to their triple product, so three non-zero vectors are coplanar if and only if their triple product is zero, For the ray $\boldsymbol{\ell}$ from the left center of projection and the ray $\mathbf{r}$ from the right center of projection to be coplanar with the baseline $\mathbf{b}$, we must have [Horn 87b,

[1] If we treat the baseline as a unit vector, its actual length becomes the unit of length for all other quantities.

90]

$$\boxed{[\mathbf{b}\ \boldsymbol{\ell}'\ \mathbf{r}] = 0} \tag{1}$$

where $\boldsymbol{\ell}'$ is the left ray rotated into the right imaging system's coordinates[2]. Using the unit quaternion $\mathring{q}$ to represent this rotation, we can write

$$\mathring{\ell}' = \mathring{q}\mathring{\ell}\mathring{q}^*, \tag{2}$$

where $\mathring{\ell}$ and $\mathring{\ell}'$ are unit quaternions with zero scalar part and vector part equal to $\boldsymbol{\ell}$ and $\boldsymbol{\ell}'$ respectively, that is,

$$\mathring{\ell} = (0, \boldsymbol{\ell}) \quad \text{and} \quad \mathring{\ell}' = (0, \boldsymbol{\ell}'). \tag{3}$$

(For use of unit quaternion notation in a related photogrammetric problem, including a discussion of numerically stable methods for converting between orthonormal matrix notation and unit quaternion notation, see [Horn 87a]. See also Appendix A and [McCarthy 90].) We can write the triple product in the form

$$t = (\mathbf{r} \times \mathbf{b}) \cdot \boldsymbol{\ell}', \tag{4}$$

or, letting $\mathring{b} = (0, \mathbf{b})$ and $\mathring{r} = (0, \mathbf{r})$,

$$t = \mathring{r}\mathring{b} \cdot \mathring{q}\mathring{\ell}\mathring{q}^*, \tag{5}$$

where we have used the fact that $\mathring{\ell}' = \mathring{q}\mathring{\ell}\mathring{q}^*$ has zero scalar part and

$$\mathring{r}\mathring{b} = (-\mathbf{r} \cdot \mathbf{b}, \mathbf{r} \times \mathbf{b}), \tag{6}$$

since both $\mathring{r}$ and $\mathring{b}$ have zero scalar parts. The triple product can now be further transformed to yield

$$t = \mathring{r}\mathring{b}\mathring{q} \cdot \mathring{q}\mathring{\ell}, \tag{7}$$

or finally[3].

$$\boxed{t = \mathring{r}\mathring{d} \cdot \mathring{q}\mathring{\ell}} \tag{8}$$

where $\mathring{d} = \mathring{b}\mathring{q}$. Note that $\mathring{d}$ is orthogonal to $\mathring{q}$, since

$$\mathring{d} \cdot \mathring{q} = \mathring{b}\mathring{q} \cdot \mathring{q} = \mathring{b} \cdot \mathring{q}\mathring{q}^* = \mathring{b} \cdot \mathring{e} = 0, \tag{9}$$

where $\mathring{e}$ is the identity with respect to quaternion multiplication[4].

The baseline can be recovered from $\mathring{d}$ using

$$\mathring{d}\mathring{q}^* = \mathring{b}\mathring{q}\mathring{q}^* = \mathring{b}\mathring{e} = \mathring{b}, \tag{10}$$

so one may as well work with the parameters $\mathring{q}$ and $\mathring{d}$, rather than $\mathring{q}$ and $\mathring{b}$, if this is convenient. Note that the resulting expression is bilinear in the unknowns, being separately linear in the components of $\mathring{q}$ and in the components of $\mathring{d}$.

[2] Here the baseline $\mathbf{b}$ is also expressed in the right imaging system's coordinates. The coplanarity conditions can, of course, be equally well expressed in the coordinates of the left imaging system.

[3] In the above a number of quaternion identities, such as $\mathring{a}\mathring{q} \cdot \mathring{b} = \mathring{a} \cdot \mathring{b}\mathring{q}^*$, have been used that can be easily checked by using the rule for quaternion multiplication in terms of the scalar and vector parts of the quaternions given in Appendix A.

[4] The identity $\mathring{e}$ has unit scalar part and zero vector part.

1.2 Symmetry in the Coplanarity Condition

We can rewrite the triple product using

$$t = \mathring{r}\mathring{d} \cdot \mathring{q}\mathring{\ell} = \mathring{r} \cdot \mathring{q}\mathring{\ell}\mathring{d}^* = \mathring{q}^*\mathring{r} \cdot \mathring{\ell}\mathring{d}^*, \tag{11}$$

and

$$t = \mathring{q}^*\mathring{r} \cdot \mathring{\ell}\mathring{d}^* = (\mathring{q}^*\mathring{r})^* \cdot (\mathring{\ell}\mathring{d}^*)^* = \mathring{r}^*\mathring{q} \cdot \mathring{d}\mathring{\ell}^*. \tag{12}$$

Finally, noting that $\mathring{\ell}^* = -\mathring{\ell}$ and $\mathring{r}^* = -\mathring{r}$, since $\mathring{r}$ and $\mathring{\ell}$ are quaternions with zero scalar parts, we obtain

$$\boxed{t = \mathring{r}\mathring{q} \cdot \mathring{d}\mathring{\ell}} \tag{13}$$

The symmetry can be seen in more detail if the dot-product for t is expanded out in terms of the scalar and vector components of $\mathring{q} = (q, \mathbf{q})$ and $\mathring{d} = (d, \mathbf{d})$:

$$(\mathbf{d} \cdot \mathbf{r})\,(\mathbf{q} \cdot \boldsymbol{\ell}) + (\mathbf{q} \cdot \mathbf{r})\,(\mathbf{d} \cdot \boldsymbol{\ell}) + (dq - \mathbf{d} \cdot \mathbf{q})\,(\boldsymbol{\ell} \cdot \mathbf{r}) + d\,[\mathbf{r}\ \mathbf{q}\ \boldsymbol{\ell}] + q\,[\mathbf{r}\ \mathbf{d}\ \boldsymbol{\ell}]. \tag{14}$$

Certain other symmetries now become apparent. If the parameters $\{\mathring{q}, \mathring{d}\}$ satisfy the coplanarity condition for corresponding sets of rays $\{\boldsymbol{\ell}_i\}$ and $\{\mathbf{r}_i\}$, then:

- The set of parameters $\{-\mathring{q}, \mathring{d}\}$ satisfy the coplanarity condition also. This has no physical significance, however, since $-\mathring{q}$ represents the same rotation as $\mathring{q}$.
- The set of parameters $\{\mathring{q}, -\mathring{d}\}$ satisfy the coplanarity condition also. This corresponds to a reversal of the baseline $\mathbf{b}$.
- The set of parameters $\{\mathring{d}, \mathring{q}\}$ satisfy the coplanarity condition also. This corresponds to the "twisted sister dual" obtained by an additional rotation of π about the baseline [Horn 87b, 90] [Krames 40].

That is, the solutions come in groups of eight related solutions.

Also note that, perhaps somewhat surprisingly, we obtain the same set of solutions if we interchange the left and right rays, since the expression for t is symmetric in $\boldsymbol{\ell}$ and $\mathbf{r}$.

2. The New Iterative Scheme

Given two corresponding sets of rays $\{\boldsymbol{\ell}_i\}$ and $\{\mathbf{r}_i\}$ (for $i = 1, 2, \ldots n$) from the left and the right imaging systems respectively, the task is to find $\mathring{q}$ and $\mathring{d}$ that minimize

$$\sum_{i=1}^{n} w_i e_i^2, \quad \text{where} \quad e_i = (\mathring{r}_i\mathring{d} \cdot \mathring{q}\mathring{\ell}_i). \tag{15}$$

subject to

$$\mathring{q} \cdot \mathring{q} = 1, \quad \mathring{d} \cdot \mathring{d} = 1, \quad \text{and} \quad \mathring{q} \cdot \mathring{d} = 0. \tag{16}$$

The weight factor is chosen according to the reliability of a particular measurement, but also depends on the ray direction. That is, the error contributions one wishes to minimize are distances in the image plane, not in the three-dimensional world [Horn 87b, 90] It can be shown that the appropriate weighting factor is

$$w_i = \frac{\|\mathbf{c}_i\|^2 \, \sigma_o^2}{[\mathbf{c}_i \; \mathbf{b} \; \mathbf{r}_i]^2 \, \|\boldsymbol{\ell}'_i\|^2 \, \sigma_{l_i}^2 + [\mathbf{c}_i \; \mathbf{b} \; \boldsymbol{\ell}'_i]^2 \, \|\mathbf{r}_i\|^2 \, \sigma_{r_i}^2}, \tag{17}$$

where $\mathbf{c}_i = \boldsymbol{\ell}'_i \times \mathbf{r}_i$ and $\sigma_{l_i}^2$ and $\sigma_{r_i}^2$ are the estimated variances of the measurement errors of the directions of rays in the left and right images respectively, while σ_o^2 is arbitrary. Proper weighting is particularly important when the fields of view is narrow, since the relative orientation problem then is often not well conditioned. Note that the weighting factor depends on the (unknown) baseline and rotation. One way of dealing with this is to treat the weights as constant during any particular step of the iteration. One may start off with unit weights and then use the current estimate of the baseline and rotation as the iteration progresses [Horn 87b].

Exact solutions are possible when there are only five pairs of rays, so the weight factors can be omitted in this case, since they do not affect the solutions (see section 3).

2.1 Iterative Adjustment

Since no closed form solution is at hand, let us see how small changes in $\mathring{q}$ and $\mathring{d}$ affect the total error. First of all, by ignoring second order terms in

$$(\mathring{q}+\delta\mathring{q})\cdot(\mathring{q}+\delta\mathring{q}) = 1, \quad (\mathring{d}+\delta\mathring{d})\cdot(\mathring{d}+\delta\mathring{d}) = 1, \quad \text{and} \quad (\mathring{q}+\delta\mathring{q})\cdot(\mathring{d}+\delta\mathring{d}) = 0 \tag{18}$$

we obtain the following constraints on the increments

$$\mathring{q}\cdot\delta\mathring{q} = 0, \quad \mathring{d}\cdot\delta\mathring{d} = 0, \quad \text{and} \quad \mathring{q}\cdot\delta\mathring{d} + \mathring{d}\cdot\delta\mathring{q} = 0. \tag{19}$$

We have to find increments $\delta\mathring{q}$ and $\delta\mathring{d}$ that minimize

$$\sum_{i=1}^{n} w_i \left(\mathring{r}_i(\mathring{d}+\delta\mathring{d})\cdot(\mathring{q}+\delta\mathring{q})\mathring{\ell}_i\right)^2, \tag{20}$$

subject to the three constraints noted. Ignoring the second order term in the dot-product (containing both $\delta\mathring{q}$ and $\delta\mathring{d}$), and introducing Lagrange multipliers, we find that we have to minimize

$$\sum_{i=1}^{n} w_i \left(\mathring{r}_i\mathring{d}\cdot\mathring{q}\mathring{\ell}_i + \mathring{r}_i\,\delta\mathring{d}\cdot\mathring{q}\mathring{\ell}_i + \mathring{r}_i\mathring{d}\cdot\delta\mathring{q}\,\mathring{\ell}_i\right)^2 + \lambda(\mathring{q}\cdot\delta\mathring{q}) + \mu(\mathring{d}\cdot\delta\mathring{d}) + \nu(\mathring{q}\cdot\delta\mathring{d} + \mathring{d}\cdot\delta\mathring{q}). \tag{21}$$

Differentiating with respect to $\delta\mathring{q}$ and $\delta\mathring{d}$ and setting the results equal to zero, we obtain

$$\begin{aligned}
&\sum_{i=1}^{n} w_i \left(\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \cdot \mathring{\mathrm{q}} \mathring{\ell}_i + \mathring{\mathrm{r}}_i \, \delta\mathring{\mathrm{d}} \cdot \mathring{\mathrm{q}} \mathring{\ell}_i + \mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{q}} \, \mathring{\ell}_i\right) \mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^* + \lambda \mathring{\mathrm{q}} + \nu \mathring{\mathrm{d}} = 0, \\
&\sum_{i=1}^{n} w_i \left(\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \cdot \mathring{\mathrm{q}} \mathring{\ell}_i + \mathring{\mathrm{r}}_i \, \delta\mathring{\mathrm{d}} \cdot \mathring{\mathrm{q}} \mathring{\ell}_i + \mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{q}} \, \mathring{\ell}_i\right) \mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i + \mu \mathring{\mathrm{d}} + \nu \mathring{\mathrm{q}} = 0,
\end{aligned} \tag{22}$$

where we may wish to note that $\mathring{\mathrm{r}}_i* = -\mathring{\mathrm{r}}_i$ and $\mathring{\ell}_i^* = -\mathring{\ell}_i$. Differentiating with respect to the Lagrangian multipliers just gives us back the original constraints

$$\mathring{\mathrm{q}} \cdot \delta\mathring{\mathrm{q}} = 0, \quad \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{d}} = 0, \quad \text{and} \quad \mathring{\mathrm{q}} \cdot \delta\mathring{\mathrm{d}} + \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{q}} = 0. \tag{23}$$

Isolating the unknowns $\delta\mathring{\mathrm{q}}$ and $\delta\mathring{\mathrm{d}}$, we obtain

$$\begin{aligned}
A \, \delta\mathring{\mathrm{q}} \; + B \, \delta\mathring{\mathrm{d}} + \lambda \mathring{\mathrm{q}} + \nu \mathring{\mathrm{d}} &= -\mathring{\mathrm{s}}, \\
B^T \delta\mathring{\mathrm{q}} + C \, \delta\mathring{\mathrm{d}} + \mu \mathring{\mathrm{d}} + \nu \mathring{\mathrm{q}} &= -\mathring{\mathrm{t}},
\end{aligned} \tag{24}$$

where

$$A = \sum_{i=1}^{n} w_i (\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^*)(\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^*)^T, \quad B = \sum_{i=1}^{n} w_i (\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^*)(\mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i)^T,$$

$$\text{and} \quad C = \sum_{i=1}^{n} w_i (\mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i)(\mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i)^T, \tag{25}$$

while

$$\mathring{\mathrm{s}} = \sum_{i=1}^{n} w_i e_i \, (\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^*) \quad \text{and} \quad \mathring{\mathrm{t}} = \sum_{i=1}^{n} w_i e_i \, (\mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i). \tag{26}$$

We also still have the three equations

$$\mathring{\mathrm{q}} \cdot \delta\mathring{\mathrm{q}} = 0, \quad \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{d}} = 0, \quad \text{and} \quad \mathring{\mathrm{q}} \cdot \delta\mathring{\mathrm{d}} + \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{q}} = 0, \tag{27}$$

all of which we can write in the matrix form

$$\begin{pmatrix} A & B & \mathring{\mathrm{q}} & 0 & \mathring{\mathrm{d}} \\ B^T & C & 0 & \mathring{\mathrm{d}} & \mathring{\mathrm{q}} \\ \mathring{\mathrm{q}}^T & 0^T & 0 & 0 & 0 \\ 0^T & \mathring{\mathrm{d}}^T & 0 & 0 & 0 \\ \mathring{\mathrm{d}}^T & \mathring{\mathrm{q}}^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta\mathring{\mathrm{q}} \\ \delta\mathring{\mathrm{d}} \\ \lambda \\ \mu \\ \nu \end{pmatrix} = - \begin{pmatrix} \mathring{\mathrm{s}} \\ \mathring{\mathrm{t}} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{28}$$

So we have a system of 11 equations in 11 unknowns, four of which are the components of $\mathring{\mathrm{q}}$, four are the components of $\mathring{\mathrm{d}}$, and three are Lagrangian multipliers.

Since we are usually not really interested in the values of the Lagrange multipliers, we may eliminate them using the conditions

$$\mathring{\mathrm{q}} \cdot \delta\mathring{\mathrm{q}} = 0, \quad \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{d}} = 0, \quad \text{and} \quad \mathring{\mathrm{q}} \cdot \delta\mathring{\mathrm{d}} + \mathring{\mathrm{d}} \cdot \delta\mathring{\mathrm{q}} = 0. \tag{29}$$

leaving us with 8 equations in 8 unknowns. But this takes some effort, while spoiling the symmetry of the normal matrix, and so may not be desirable.

Note that the upper left 8 × 8 sub-matrix is the weighted sum of dyadic products

$$\sum_{i=1}^{n} w_i \vec{c}_i \vec{c}_i^{\,T}, \tag{30}$$

where the eight component vector $\vec{c}_i$ is given by

$$\vec{c}_i = \begin{pmatrix} \mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^* \\ \mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i \end{pmatrix} = - \begin{pmatrix} \mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i \\ \mathring{\mathrm{r}}_i \mathring{\mathrm{q}} \mathring{\ell}_i \end{pmatrix}. \tag{31}$$

Also note that the eight non-zero components of the right-hand side vector are given by the weighted sum

$$\sum_{i=1}^{n} w_i e_i \, \vec{c}_i \quad \text{where} \quad e_i = (\mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \cdot \mathring{\mathrm{q}} \mathring{\ell}_i). \tag{32}$$

For computational purposes it may be further helpful to note that

$$e_i = \mathring{\mathrm{r}}_i^* \mathring{\mathrm{q}} \mathring{\ell}_i \cdot \mathring{\mathrm{d}} = \mathring{\mathrm{r}}_i \mathring{\mathrm{d}} \mathring{\ell}_i^* \cdot \mathring{\mathrm{q}}. \tag{33}$$

A step in the iterative algorithm consist of computing the coefficient matrix above, as well as the right hand side vector, solving for $\delta \mathring{\mathrm{d}}$ and $\delta \mathring{\mathrm{q}}$, and then updating $\mathring{\mathrm{d}}$ and $\mathring{\mathrm{q}}$ accordingly.

2.2 Keeping the Quaternions Orthogonal

In practice, the updated quaternions

$$\mathring{\mathrm{q}}' = \mathring{\mathrm{q}} + \delta \mathring{\mathrm{q}} \quad \text{and} \quad \mathring{\mathrm{d}}' = \mathring{\mathrm{d}} + \delta \mathring{\mathrm{d}} \tag{34}$$

will not be exactly orthogonal, even if $\mathring{\mathrm{d}}$ and $\mathring{\mathrm{q}}$ where, because of the finite step size of the increment. It is therefore important to adjust the new values to make them more nearly orthogonal. The smallest adjustment that will achieve this is obtained by finding k such that $\mathring{\mathrm{q}}'' \cdot \mathring{\mathrm{d}}'' = 0$, where

$$\mathring{\mathrm{q}}'' = \mathring{\mathrm{q}}' + k \, \mathring{\mathrm{d}}' \quad \text{and} \quad \mathring{\mathrm{d}}'' = \mathring{\mathrm{d}}' + k \, \mathring{\mathrm{q}}'. \tag{35}$$

This leads to a quadratic equation for k. The solution for k has the term $(\mathring{\mathrm{q}}' \cdot \mathring{\mathrm{d}}')$ in the denominator, and so is numerically unstable when $\mathring{\mathrm{q}}'$ and $\mathring{\mathrm{d}}'$ are already nearly orthogonal. If instead we ignore the term in k^2, we obtain

$$k \approx -\frac{\mathring{\mathrm{q}}' \cdot \mathring{\mathrm{d}}'}{\mathring{\mathrm{q}}' \cdot \mathring{\mathrm{q}}' + \mathring{\mathrm{d}}' \cdot \mathring{\mathrm{d}}'} \approx -\frac{1}{2}(\mathring{\mathrm{q}}' \cdot \mathring{\mathrm{d}}'). \tag{36}$$

While an adjustment based on this value of k will not make the two quaternions exactly orthogonal, it will insure that they converge to orthogonal values after a number of steps of the iteration. It is, of course, a simple matter to adjust the quaternions to have unit magnitude. This should be done *after* the adjustment to obtain more nearly orthogonal vectors.

Sometimes when the starting values are far from a minimum, a large adjustment suggested by the above algorithm may make matters worse rather than better. As an added refinement, one can compare the error after the adjustment with that before, and take only half the step if the error has increased. If the error with the smaller step is also larger than the initial error, the step size can again be halved. Repeated halving of the step size in this fashion will normally only occur when one is very close to the solution and the algorithm is unable to reduce the error due to limitations of computer arithmetic. This condition may be used as a termination test for the iteration.

Typically a solution is found to single precision after fewer than ten iterations. In some cases convergence is slow, however, particularly when the initial guess is near a saddle point. To avoid wasting time in this case, one may wish to insist that the error after the adjustment not merely be smaller, but that it be smaller by a reasonable fraction, say 1% of the old error. The iteration is abandoned if it is not improving the solution at least this much. The minimum that might have been reached after a long computation will almost certainly be reached from some other starting value, so nothing is lost by abandoning a particular solution path.

2.3 Starting Values

To find all local minima, and so be in a position to determine the global minimum, a number of different starting values for the orientation $\mathring{q}$ are needed. These can be generated:

- using the elements of a finite rotation group [Horn 87b, 90].
- by some other systematic sampling of a unit hemisphere, or
- at random.

Points on the unit hemisphere (in four dimensions) may be generated systematically using

$$\mathring{q} = (\cos\alpha\cos\beta\cos\gamma,\ \cos\alpha\cos\beta\sin\gamma,\ \cos\alpha\sin\beta,\ \sin\alpha).$$

If the proper ranges of α, β, and γ are divided evenly, an uneven sampling of the hemisphere results, which is wasteful, in that in order to achieve a given minimum sampling rate in some areas, other areas need to be sampled much more finely. To obtain roughly even sampling of the hemispherical surface, discrete sampling in each of the three variables can be made dependent on the other variables. While sampling evenly in α, one should sample β at a rate proportional to $\cos\alpha$, and sample γ at a rate proportional to $\cos\alpha\cos\beta$. This is analogous to sampling the unit sphere using longitude and latitude, where, to avoid oversampling near the poles, one should sample along parallels at a rate that is proportional to the cosine of the latitude.

The number of starting values needed is greatly reduced if it is noted that each solution $\{\mathring{q}, \mathring{d}\}$ that is found belongs to a set of eight related solutions obtained by changing the signs of $\mathring{q}$ and $\mathring{d}$, and by interchanging $\mathring{q}$ and $\mathring{d}$, as discussed earlier. Typically all solutions are found after trying a few dozen initial guesses. If a particular solution has a small basin of attraction in parameter space, it will typically still be found, since it is very unlikely that all seven of the solutions related to it *also* have small basins of attraction.

While it might be expected that random sampling should be less efficient, in that a larger number of samples are needed to ensure that the largest gaps between samples are as small as they are between samples generated by some systematic method, it typically appears not to take a larger number of starting values to find all the solutions from random starting points. This simplifies the algorithm. Typically all solutions are found from fewer than thirty or so random starting values. Occasionally one solution will be missed. But in this case the number of solutions found is an odd an multiple of four, and the search can be extended when this is noted.

2.4 Finding $\mathring{d}$ given $\mathring{q}$ (and *vice versa*)

If either of the unit quaternions $\mathring{q}$ or $\mathring{d}$ is known, it is possible to find a best fit value for the other. This is useful when setting up starting values, since it means that one only need explore the unit sphere in four space for one of the two sets of parameters. Since the total error term is completely symmetric in $\mathring{q}$ and $\mathring{d}$, we need only explore one of the two cases. Suppose for concreteness that $\mathring{q}$ is known and we are to find the best fit value for $\mathring{d}$.

We can look for the $\mathring{d}$ that minimizes

$$\sum_{i=1}^{n} w_i(\mathring{r}_i\mathring{d}\cdot\mathring{q}\mathring{\ell}_i)^2, \tag{37}$$

subject to

$$\mathring{d}\cdot\mathring{d} = 1 \quad \text{and} \quad \mathring{q}\cdot\mathring{d} = 0. \tag{38}$$

Reversing our argument at the beginning regarding the form of the error term, we write the above in the form

$$\sum_{i=1}^{n} w_i(\mathring{\ell}'_i\mathring{r}_i\cdot\mathring{b})^2, \tag{39}$$

where $\mathring{b} = \mathring{d}\mathring{q}^*$ and $\mathring{\ell}'_i = \mathring{q}\mathring{\ell}_i\mathring{q}^*$. We know that $\mathring{\ell}'_i$ and $\mathring{r}_i$ have zero scalar part. But $\mathring{b}$ also has zero scalar part since

$$\mathring{b}\cdot\mathring{e} = \mathring{d}\mathring{q}^*\cdot\mathring{e} = \mathring{d}\cdot\mathring{q} = 0. \tag{40}$$

So the above can be written

$$\sum_{i=1}^{n} w_i [\mathbf{b} \boldsymbol{\ell}'_i \; \mathbf{r}_i]^2, \tag{41}$$

where $\mathring{\mathrm{b}} = (0, \mathbf{b})$, and $\mathring{\ell}'_i = (0, \boldsymbol{\ell}'_i)$. Now

$$\mathbf{b} \cdot \mathbf{b} = \mathring{\mathrm{b}} \cdot \mathring{\mathrm{b}} = (\mathring{\mathrm{d}}\mathring{\mathrm{q}}^*)^T \mathring{\mathrm{d}}\mathring{\mathrm{q}}^* = \mathring{\mathrm{d}} \cdot \mathring{\mathrm{d}}\, \mathring{\mathrm{q}}^* \mathring{\mathrm{q}} = \mathring{\mathrm{d}} \cdot \mathring{\mathrm{d}} = 1. \tag{42}$$

So the condition that $\mathring{\mathrm{d}}$ be a unit quaternion is equivalent to the condition that $\mathbf{b}$ is a unit vector. So we are trying to minimize

$$\mathbf{b}^T \left(\sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{c}_i^T \right) \mathbf{b} \tag{43}$$

where $\mathbf{c}_i = \boldsymbol{\ell}'_i \times \mathbf{r}_i$, subject to $\mathbf{b} \cdot \mathbf{b} = 1$. The solution is the eigenvector of the 3×3 matrix associated with its smallest eigenvalue [Horn 87b, 90] (see also the discussion of Raleigh's quotient in [Korn & Korn 68]). From $\mathbf{b}$ we can recover $\mathring{\mathrm{d}}$ using $\mathring{\mathrm{d}} = \mathring{\mathrm{b}}\mathring{\mathrm{q}}$, where $\mathring{\mathrm{b}} = (0, \mathbf{b})$.

It has been found experimentally, perhaps somewhat surprisingly, that one can actually just pick a random initial value for $\mathring{\mathrm{d}}$. Convergence to machine precision is on average delayed by less than one step compared to the number of steps needed when the method described here is used to find an optimal initial value for $\mathring{\mathrm{d}}$. This simplifies the algorithm.

3. Five Ray Pairs

The minimum number of ray pairs that yield a finite number of solutions is five, since each pairing of rays yields one constraint, and there are five unknowns. There are five degrees of freedom, because there are three constraints on the eight components of $\mathring{\mathrm{q}}$ and $\mathring{\mathrm{d}}$—the two quaternions have to be orthogonal and of unit magnitude. With five rays pairs exact solutions are possible, that is, solutions that satisfy the coplanarity condition exactly. In practice, if at all possible, one uses more than five ray pairs in order to achieve higher accuracy and avoid ambiguity. Nevertheless, this minimal case has attracted some attention and is worth discussing.

The question of how many solutions there may be when five ray pairs are given has been long debated. Since each ray pair yields a homogeneous second-degree polynomial in the unknown components of $\mathring{\mathrm{q}}$ and $\mathring{\mathrm{d}}$, we see right away by Bézout's theorem that there can be at most $2^5 = 32$ solutions (ignoring sign changes of $\mathring{\mathrm{q}}$ and $\mathring{\mathrm{d}}$). Kruppa showed long ago, however, that there can actually be no more than 22 solutions [Kruppa 13]. More recently, it has been observed experimentally that there appear to never be more than twenty solutions, that these solutions generally come in groups of four, and that sets of ray pairs can be found that yield no solutions, or as many as twenty [Horn 87a, 90]. Proofs that there can be no more than twenty solutions have recently been given by [Demazure 88] [Faugeras & Maybank 89] [Netravali *et al.* 89]. But these proofs are very complex and use advanced concepts from projective geometry and algebraic geometry.

We can show more simply that there can be no more than twenty solutions by noting that the equations are bi-linear, that is, separately linear in the components of $\mathring{q}$ and the components of $\mathring{d}$. This means that the equations derived from the coplanarity conditions are actually 2-homogeneous (see Appendix B). The number of solutions of a system of m-homogeneous equations is less than that of a general homogeneous system of equations of the same degree. In our case, we have five equations that are linear and homogeneous in each of two sets of four variables, so the maximum possible number of solutions is given by

$$\binom{(8-2)}{(4-1)} = \frac{6!}{3!\,3!} = 20.$$

Methods have been developed for tracking the paths of roots as one system of polynomials is continuously transformed into another [Morgan 87]. These methods can be used here to track the roots from a special system of equations with the same degree that can be solved explicitly, as it is transformed into the system of equations equations arising from the given ray pairs (see Appendix C). One can even exploit the symmetry of the equations in $\mathring{q}$ and $\mathring{d}$ so that one only needs to track 10 roots, not 20.

Preliminary experiments with this method suggest, however, that the iterative method described earlier, designed for the more general least squares problem when more than five ray pairs are given, is much faster and also more reliable. One problem with continuation methods is that, while in theory paths of roots should never cross, in practice they often come close enough to allow "path jumping," unless the path is followed with impractically tight tolerances.

4. Conclusions

An elegant new iterative method for solving the least squares problem of relative orientation has been described. The utilitity of unit quaternions for representing rotations in three-dimnesional space has once again been demonstrated. A new short proof has been given that there can be at most twenty solutions of the relative orientation problem when only five ray pairs are given. In this special case continuation methods can (at least theoretically) find all of the solutions.

5. Acknowledgments

The author would like to thank Professor John Canny, Professor Bernard Roth, Doctor Alexander Morgan, Shawney Harpeet, and Mark Snyder for helpful comments. Olivier Faugeras and Steve Maybank provided references to early work on photogrammetry, and Jason Grant pointed out several typos.

6. References

Demazure, M. (1988) "Sur Deux Problemes de Reconstruction," INRIA Report 882 Institut National de Recherche en Informatique et en Automatique, Domaine de Volcueau, Rocquencourt, Les Chesnay, Cedex, France.

Finsterwalder, S. (1899) "Die geometrischen Grundlagen der Photogrametrie," *Jahresbericht Deutscher Mathematik*, Vol. 6, pp. 1–44.

Hauck, G. (1883) "Neue Konstruktionen der Perspektive und Photogrammetrie," *Crelle J. f. Math.*, pp. 1–35.

Horn, B.K.P. (1987a) "Closed-Form Solution of Absolute Orientation using Unit Quaternions," *Journal of the Optical Society of America A*, Vol. 4, No. 4, pp. 629–642, April.

Horn, B.K.P. (1987b) "Relative Orientation," Memo 994, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts. November.

Horn, B.K.P. (1990) "Relative Orientation," *International Journal of Computer Vision*, Vol. 4, No. 1, pp. 59–78.

Korn, G.A. & T.M. Korn (1968) *Mathematical Handbook for Scientists and Engineers*, 2-nd edition, McGraw-Hill, New York, NY.

Krames, J. (1940–41) "Zur Ermittlung eines Objektes aus zwei Perspektiven. (Ein Beitrag zur Theorie der 'gefährlichen Örter'.)," *Monatshefte für Mathematik und Physik*, Vol. 49, pp. 327–354.

Kruppa, E. (1913) *Sitzgsber. Akade. Wien*, Math.-Nat., IIa, No. 122, pp. 1939–1948.

Faugeras, O.D. & S. Maybank (1989) "Motion from Point Matches: Multiplicity of Solutions," *Proceedings of IEEE Workshop on Motion Vision*, Irvine, CA, March 20–22.

McCarthy, J. (1990) "Introduction to Theoretical Kinematics," MIT Press, Cambridge, Massachusetts.

Morgan, A.P. (1987) *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*, Prentice-Hall, Englewood Cliffs NJ.

Morgan, A.P. (1989) "Polynomial Continuation," *Impacts of Recent Advances on Operations Research*, Sharda, R., B.L. Golden, E. Wasil, O. Balci, & W. Stewart (eds.), Elsevier Science Publishing Co., pp. 101–113.

Morgan, A.P. & A. Sommese (1987a) "A Homotopy for Solving General Polynomial Systems That Respects m-Homogeneous Structures," *Applied Mathematics and Computation*, Vol. 24, pp. 101–113.

Morgan A.P. & A. Sommese (1987b) "Computing All Solutions to Polynomial Systems Using Homotopy Continuation," *Applied Mathematics and Computation*, Vol 24, pp. 115–138

Morgan, A.P & A.J. Sommese (1989) "Coefficient-Parameter Polynomial Continuation," *Applied Mathematics and Computation*, Vol. 29, pp. 123–160.

Netravali, A.N., T.S. Huang, A.S. Krishnakumar & R.J. Holt (1989) "Alegbraic Methods in 3-D Motion Estimation from Two-View Point Correspondences," *International Journal of Imaging Systems and Technology*, Vol. 1, pp. 78–99.

Raghavan, M. & B. Roth (1989) "Kinematic Analysis of the 6R Manipulator of General Geometry," ISSR, Tokyo, pp. 314–320.

Wampler, C.W., A.P. Morgan & A.J. Sommese (1988) "Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics," Research Report GMR-6372, General Motors Research Laboratories, Warren, Michigan, August. To appear in *ASME Journal of Mechanisms, Transmissions, and Automation in Design.*

A. Quaternion Products and Rotation in 3-D

It is often convenient to consider quaternions as composed of a scalar and a vector part:

$$\mathring{a} = (a, \mathbf{a}). \tag{44}$$

The conjugate of a quaternion is the quaternion with the vector part negated:

$$\mathring{a}^* = (a, -\mathbf{a}). \tag{45}$$

The dot-product of two quaternions is a scalar given by

$$\mathring{a} \cdot \mathring{b} = (a, \mathbf{a}) \cdot (b, \mathbf{b}) = ab + \mathbf{a} \cdot \mathbf{b}. \tag{46}$$

The norm of a quaternion is just the square root of the dot-product of the quaternion with itself:

$$\|\mathring{a}\| = \sqrt{\mathring{a} \cdot \mathring{a}}. \tag{47}$$

A unit quaternion is a quaternion of unit norm.

The quaternion product is defined by the relation

$$\mathring{a}\mathring{b} = (a, \mathbf{a})\,(b, \mathbf{b}) = (ab - \mathbf{a} \cdot \mathbf{b},\ a\,\mathbf{b} + b\,\mathbf{a} + \mathbf{a} \times \mathbf{b}). \tag{48}$$

The appearance of the cross product in the result alerts us to the fact that quaternion multiplication is not commutative. Quaternion multiplication is associative, however. It is easy to see that the identity with respect to multiplication is

$$\mathring{e} = (1, \mathbf{0}), \tag{49}$$

where **0** is the vector whose components are all zero. Note that

$$\mathring{a}\mathring{a}^* = (a, \mathbf{a})\,(a, -\mathbf{a}) = (a^2 + \mathbf{a} \cdot \mathbf{a},\ \mathbf{0}) = (\mathring{a} \cdot \mathring{a})\,\mathring{e}, \tag{50}$$

so that a quaternion with non-zero norm has an inverse,

$$\mathring{a}^{-1} = \mathring{a}^{*}/\|\mathring{a}\|^{2}, \tag{51}$$

and the inverse of a unit quaternion is just its conjugate.

Using the definition given above of the quaternion product, it is easy to show that

$$(\mathring{a}\mathring{q}) \cdot (\mathring{b}\mathring{q}) = (\mathring{a} \cdot \mathring{b})\,(\mathring{q} \cdot \mathring{q}). \tag{52}$$

We conclude that the operation of multiplying by a unit quaternion preserves dot-products. We also obtain as a special case

$$(\mathring{a}\mathring{b}) \cdot (\mathring{a}\mathring{b}) = (\mathring{a} \cdot \mathring{a})\,(\mathring{b} \cdot \mathring{b}), \tag{53}$$

thus the norm of a product is the product of the norms. Using these results, we can also see that

$$(\mathring{a}\mathring{q}) \cdot \mathring{b} = \mathring{a} \cdot (\mathring{b}\mathring{q}^{*}). \tag{54}$$

Scalars can be represented by quaternions with zero vector part, while vectors can be represented by quaternions with zero scalar part. If $\mathring{r}$ is a quaternion with zero scalar part, then

$$\mathring{r}^{*} = -\mathring{r}. \tag{55}$$

If $\mathring{r}$ and $\mathring{s}$ are quaternions with zero scalar part then

$$\mathring{r} \cdot \mathring{s} = \mathbf{r} \cdot \mathbf{s}, \tag{56}$$

and

$$\mathring{r}\mathring{s} = (-\mathbf{r} \cdot \mathbf{s},\, \mathbf{r} \times \mathbf{s}) = (\mathring{s}\mathring{r})^{*}. \tag{57}$$

Finally, if $\mathring{r}$, $\mathring{s}$ and $\mathring{t}$ are quaternions with zero scalar part, then

$$(\mathring{r}\mathring{s}) \cdot \mathring{t} = \mathring{r} \cdot (\mathring{s}\mathring{t}) = [\mathbf{r}\ \mathbf{s}\ \mathbf{t}], \tag{58}$$

To represent rotations in three-dimensional space, we need an operation that maps quaternions with zero scalar part into quaternions with zero scalar part. The operation

$$\mathring{r}' = \mathring{q}\mathring{r}\mathring{q}^{*} \tag{59}$$

multiplies the scalar part by $(\mathring{q} \cdot \mathring{q})$, that is

$$r' = (q^{2} + \mathbf{q} \cdot \mathbf{q})\, r, \tag{60}$$

so that if $\mathring{r}$ has zero scalar part, so will $\mathring{r}'$. As for the vector part, we can write

$$\mathbf{r}' = (q^{2} + \mathbf{q} \cdot \mathbf{q})\,\mathbf{r} + 2\,q\,(\mathbf{q} \times \mathbf{r}) + 2\,\mathbf{q} \times (\mathbf{q} \times \mathbf{r}). \tag{61}$$

If $\mathring{q}$ is a unit quaternion, then the above simplifies further, and $\mathring{r}'$ actually has the same magnitude as $\mathring{r}$, that is, $(\mathring{r}' \cdot \mathring{r}') = (\mathring{r} \cdot \mathring{r})$.

If $\mathring{s}$ is a second quaternion with zero scalar part, then

$$\mathring{r}' \cdot \mathring{s}' = (\mathring{q}\mathring{r}\mathring{q}^{*}) \cdot (\mathring{q}\mathring{s}\mathring{q}^{*}) = \mathring{r} \cdot \mathring{s}. \tag{62}$$

Thus dot-products are preserved by the operation. The signs of triple products are also preserved, since

$$(\mathring{r}'\mathring{s}') \cdot \mathring{t}' = (\mathring{r}\mathring{s}) \cdot \mathring{t}. \tag{63}$$

Since length of vectors, angles between them, and the handedness of triads are preserved, we conclude that $\mathring{r}' = \mathring{q}\mathring{r}\mathring{q}^*$ corresponds to a proper rotation of the vector $\mathbf{r}$ into the vector $\mathbf{r}'$. We next determine what this rotation is.

From

$$(q, \mathbf{q})\,(0, \mathbf{q})\,(q, -\mathbf{q}) = (q^2 + \mathbf{q} \cdot \mathbf{q})\,(0, \mathbf{q}) = (0, \mathbf{q}) \tag{64}$$

we conclude that $\mathbf{q}$ is parallel to the axis of rotation. Now suppose that $\mathbf{r}$ is a unit vector perpendicular to the axis of rotation, that is, $\mathbf{r} \cdot \mathbf{r} = 1$ and $\mathbf{r} \cdot \mathbf{q} = 0$. The cosine of the angle of rotation is then given by the dot-product of $\mathbf{r}$ and $\mathbf{r}'$. Then, if $\mathring{r}' = \mathring{q}\mathring{r}\mathring{q}^*$, we have

$$\mathbf{r}' \cdot \mathbf{r} = \mathring{r}' \cdot \mathring{r} = (\mathring{q}\mathring{r}) \cdot (\mathring{r}\mathring{q}) \tag{65}$$

or

$$\cos\theta = q^2 - \mathbf{q} \cdot \mathbf{q}, \tag{66}$$

where θ is the angle of rotation. The sine of the angle of rotation is given by the triple product of $\mathbf{r}'$, $\mathbf{r}$ and a unit vector in the direction of the axis of rotation. Now

$$[\mathbf{r}'\ \mathbf{r}\ \mathbf{q}] = (\mathring{r}'\mathring{r}) \cdot (0, \mathbf{q}) = 2q\,(\mathbf{q} \cdot \mathbf{q}), \tag{67}$$

so

$$\sin\theta = 2q\|\mathbf{q}\|. \tag{68}$$

Finally, using $q^2 + \mathbf{q} \cdot \mathbf{q} = 1$, and some trigonometric identities for multiple angles, we obtain $q^2 = (\cos\theta + 1)/2$ or

$$q = \cos(\theta/2) \quad \text{and} \quad \mathbf{q} = \hat{\boldsymbol{\omega}}\,\sin(\theta/2), \tag{69}$$

where $\hat{\boldsymbol{\omega}}$ is a unit vector parallel to the axis of rotation.

Thus a rotation about an axis through the origin parallel to the unit vector $\hat{\boldsymbol{\omega}}$ can be represented by the unit quaternion

$$\mathring{q} = \left(\cos\frac{\theta}{2},\ \hat{\boldsymbol{\omega}}\sin\frac{\theta}{2}\right). \tag{70}$$

Note, however, that $-\mathring{q}$ represents the same rotation, since

$$(-\mathring{q})\mathring{r}(-\mathring{q})^* = \mathring{q}\mathring{r}\mathring{q}^*. \tag{71}$$

Thus the space of proper rotations in three dimensional space is isomorphic to the unit sphere in four dimensions, SO_3, with anti-podal points identified. Alternatively, we can identify it with the projective space P_3.

B. Systems of m-Homogeneous Equations

A polynomial is *homogeneous* in a set of variables if, and only if, it is the sum of terms of the same degree in these variables. Any non-zero multiple of a solution of a homogeneous system of equations is clearly also a solution, since each term in the polynomial is multiplied by the same power of the constant multiplier. To obtain a unique solution we have to impose an additional (linear, non-homogeneous) constraint. Given this extra degree of freedom, a homogeneous system of equations in n variables need typically consist of only $(n-1)$ equations in order to yield a finite number of solutions (up to a constant multiplier). In general, the maximum number of solutions that a system of homogeneous equations can have is equal to the product of the degrees of the equations (Bézout's theorem). Most systems of equations actually attain this maximal number of (possibly complex) solutions.

B.1 Homogeneous Equations with Special Structure

When the system of equations has some special structure, however, the maximum possible number of solutions may be lower than indicated above. Consider, for example, the pair of homogeneous second-degree equations

$$\begin{aligned} a\,xu + b\,xv + c\,yu + d\,yv &= 0, \\ e\,xu + f\,xv + g\,yu + h\,yv &= 0, \end{aligned} \tag{72}$$

in the variables x, y, and u, v. We can easily eliminate the term in xu and so obtain

$$(eb - af)\,xv + (ec - ag)\,yu + (ed - ah)\,yv = 0. \tag{73}$$

Using this to substitute for u in the first equation leads to

$$(eb - af)\,x^2 + \big((bg - fc) + (ed - ah)\big)\,xy + (gd - ch)\,y^2 = 0. \tag{74}$$

This is a homogeneous quadratic equation and so has only two solutions (up to a constant multiplier). Thus the original pair of equations has fewer solutions than the four predicted by multiplication of the degrees.

What is special about this particular system of equations is that the polynomials are separately homogeneous in the two variables x and y, and in the two variables u and v. That is, if we treat u and v as constants, then we have a pair of equations that is homogeneous in x and y (and vice versa). This means, amongst other things, that we can multiply x and y in a solution by one non-zero constant and u and v by another non-zero constant and still have a solution. That is, to obtain a unique solution we would have to introduce two additional (linear, non-homogeneous) constraints. It is because of these two degrees of freedom that we require only two equations, instead of the expected three, in order to constrain the problem enough to obtain a finite number of solutions (up to constant multipliers).

The above set of equations is said to be 2-*homogeneous*. An equation is m-*homogeneous* if we can partition the set of variables into m subsets, such that the equation is homogeneous in each of these subsets separately (when the other variables are treated as constants). The largest possible number of roots of a system of m-homogeneous equations is less than the largest possible number of roots of a general system of homogeneous equations of the same degree.

B.2 Linear 2-Homogeneous Equations

Consider, for example, a system $\vec{\ }$ of equations of $(n+m-2)$ equations that is linear in two sets of variables $\{x_i\}$ and $\{y_j\}$, where $i = 0, 1 \ldots (n-1)$ and $j = 0, 1 \ldots (m-1)$. For a start, let us focus on a very special case of this, where each of the equations happens to have the simple form

$$\begin{aligned}&(a_{0,k}x_0 + a_{1,k}x_1 + \ldots + a_{n-1,k}x_{n-1})\\ &\qquad\qquad \times (b_{0,k}y_0 + b_{1,k}y_1 + \ldots + b_{m-1,k}y_{m-1}) = 0,\end{aligned} \tag{75}$$

for $k = 0, 1 \ldots (n+m-3)$, or

$$(\mathbf{a}_k \cdot \mathbf{x})(\mathbf{b}_k \cdot \mathbf{y}) = 0 \tag{76}$$

for short, where the variables in the two subsets are the components of the vectors $\mathbf{x}$ and $\mathbf{y}$, and the two sets of coefficients are the components of the vectors $\mathbf{a}_k$ and $\mathbf{b}_k$.

Clearly for $\mathbf{x}$ and $\mathbf{y}$ to be a solution of this special system of equations $\overline{\ }$, we must have either $(\mathbf{a}_k \cdot \mathbf{x}) = 0$ or $(\mathbf{b}_k \cdot \mathbf{y}) = 0$ for each $k = 0, 1 \ldots (n+m-3)$. Suppose that we partition the system of equations into two subsets, one of size $(n-1)$ and the other of size $(m-1)$. Consider the $(n-1)$ equations $(\mathbf{a}_k \cdot \mathbf{x}) = 0$ in the first subset. This is a set of linear homogeneous equations with one fewer equations than there are variables. Generally this subset of equations will have a unique solution for $\mathbf{x}$ (up to a constant multiplier). Similarly, the $(m-1)$ equations $(\mathbf{b}_k \cdot \mathbf{y}) = 0$ in the second subset will have a unique solution for $\mathbf{y}$ (up to a constant multiplier). The resulting values of $\mathbf{x}$ and $\mathbf{y}$ are clearly solutions of the original system of equations, and there are no other solutions of the original system of equations.

We conclude that the special system of equations has a number of solutions equal to the number of ways of partitioning the set of variables in the indicated manner, namely

$$\binom{n+m-2}{n-1} = \binom{n+m-2}{m-1} = \frac{(n+m-2)!}{(n-1)!\,(m-1)!} \tag{77}$$

This typically is much less than the number of solutions of a general homogeneous system of second degree equations.

Now suppose that we have a system of equations $\vec{\ }$ that, while linear in two sets of variables, does not have the special form above. We can always write these equations in the form

$$\mathbf{x}^T M_k\, \mathbf{y} = 0, \tag{78}$$

for $(n + m - 2)$ matrices M_k, each with n rows and m columns. What is the largest number of solutions that such a system of equations can have? We can form linear combinations of this system of equations and a system of equations that do have the special form given above. The result can be written

$$\mathbf{x}^T \left(\lambda M_k + c\,(1 - \lambda)\,\mathbf{a}_k \mathbf{b}_k^T\right) \mathbf{y} = 0. \tag{79}$$

where c is an arbitrary (complex) number. Now this system has the roots of the special set of equations $^{-}$ when $\lambda = 0$, while it has the roots of the more general system of equations $^{\rightarrow}$ when $\lambda = 1$.

We can follow the roots of the combined system as we continuously vary the parameter λ. Perhaps somewhat surprisingly, the paths connect the roots of one system with the roots of the other system. None of the paths can "curve back," or merge, or diverge to infinity. So, in general, the number of roots of the more general system of equations is the same as that of the special systems of equations. The proof requires advanced concepts from modern algebraic geometry and will not be given here [Morgan 87, 89] [Morgan & Sommese 87a, 87b, 89] [Wampler, Morgan & Sommese 88]

B.3 Linear m-Homogeneous Equations

The above analysis can be easily extended to systems of equations that are linear in m sets of variables rather than just 2. A special set of equation can be set up, much as above, where each polynomial is the product of terms linear in each of the subsets of variables. This special set of $(n_0 + n_1 + \ldots + n_{m-1} - m)$ equations may be partitioned into subsets of size $(n_0 - 1)$, $(n_1 - 1)$ $\ldots$ $(n_{m-1} - 1)$. The first subset is used to solve for the n_0 variables of the first subset of variables, the second subset for the n_1 variables of the second subset of variables and so on. Since each subset of equations is linear in one subset of the variables (and does not contain any of the others), one obtains exactly one solution (up to constant multipliers). The number of solutions of the special set of equations is equal to the number of possible ways of partitioning the set of variables in the indicated manner, namely

$$\frac{(n_0 + n_1 + \ldots + n_{m-1})!}{(n_0 - 1)!\,(n_1 - 1)!\,\ldots\,(n_{m-1} - 1)!}. \tag{80}$$

which is much less than the number of solutions of a general homogeneous system of m-th degree equations.

Again, it turns out that the number of solutions of the more general set of equations is equal to the number of solutions of the special set of equations (and that the solutions of the general set may be found by following the solutions as one system of equations is deformed into the other).

The above analysis can be extended also to deal with systems of m-homogeneous equations that are of higher degree in the various subsets of variables. The same trick is used to partition the equations of the special system, but now the resulting sets of equations are no longer linear, so there will be more than one solution. Let us suppose, first of all, that all the equations have the same degrees in each of the subsets of variables. Suppose that each equation has degree l_k in the k-th set of variables. Then each partitioning leads to

$$l_0^{n_0-1}\, l_1^{n_1-1} \ldots l_{m-1}^{n_{m-1}-1} \tag{81}$$

solutions (by Bézout's theorem). So the total number of solutions is just the product of this quantity and the expression given above for the linear case.

Counting the total number of solutions becomes a bit harder when the equations are not all of the same degree in a particular subset of the variables, for then the number of solutions obtained for different partitions is different. The reader is here referred to [Morgan 87, 89] [Morgan & Sommese 87a, 87b, 89] [Wampler, Morgan & Sommese 88] for details.

C. Continuation Methods

The results discussed above can be used to determine the maximum number of solutions of an m-homogeneous system of equations. They can also be used to find these solutions using continuation methods. Let us write the system of equations we wish to solve in the form $\mathbf{f}(\mathbf{x}) = 0$. There typically is no closed-form method for finding the solution of this system. But suppose that by changing some parameters we can simplify the system of equations to the point were its solutions *can* be found directly. Of course, these will be solutions of the 'deformed' system, not the one we originally desired to solve. The idea now is to track these solutions of the 'deformed' system as it is incrementally changed back into the original form. If the incremental changes are small enough, then it is possible to get good estimates of the solutions of the next version of the system by starting with the solutions of the present one. If we are fortunate, then none of the solutions lead to 'dead-ends' where the new system has no solutions near solutions of the present system, and no new solutions can appear that are not near solutions of the present system. This is the intuitive motivation for the process to be described in more detail now.

We construct a system $\mathbf{g}(\mathbf{x}) = 0$ of equations of the same degree in the same set of variables $\mathbf{x}$, in the special form indicated in the previous sections (The coefficients occurring in these equations should be chosen at random, in order to reduce the possibility of this system having a special structure that may lead to a reduced number of solutions). Determine all of the ways of partitioning this set of equations into subsets of size one less than the number of variables in each of the m groups. Find the roots of each subset of equations extracted. This yields all of the solutions of the system $\mathbf{g}(\mathbf{x}) = 0$.

Note that to obtain unique solutions (without the constant multiplier ambiguity) we have to adjoin to the given system of equations m linear non-homogeneous equations, one in each subset of the variables (The coefficients occurring in these equations should also be chosen at random, in order to reduce the possibility of the resulting system of equations having a solution at infinity). The added linear equations can be used to solve for one of the variables in terms of the others, thus allowing this variable to be eliminated from the other equations. The result is a system of non-homogeneous equations of the same degree as the original equations, but with one fewer unknowns.

Now trace these solutions as λ is varied from 0 to 1 in

$$\lambda \mathbf{f}(\mathbf{x}) + c\,(1-\lambda)\,\mathbf{g}(\mathbf{x}) = 0,$$

or $\mathbf{h}(\mathbf{x};\lambda) = 0$ for short. This can be done by taking a small step $\delta\lambda$ in lambda and solving for the increment $\delta\mathbf{x}$ in

$$\frac{d\mathbf{h}}{d\lambda}\,\delta\lambda + \frac{d\mathbf{h}}{d\mathbf{x}}\,\delta\mathbf{x} = 0, \tag{82}$$

where $J = (d\mathbf{h}/d\mathbf{x})$ is the Jacobian of $\mathbf{h}$ with respect to $\mathbf{x}$. The updated solutions $\mathbf{x}' = \mathbf{x} + \delta\mathbf{x}$ will not be exact if we are taking finite steps, so one needs to use Newton's method to improve their accuracy. That is, we need to find an adjustment $\delta\mathbf{x}$ such that $\mathbf{h}(\mathbf{x} + \delta\mathbf{x}) = 0$, or

$$\mathbf{h}(\mathbf{x}) + \frac{d\mathbf{h}}{d\mathbf{x}}\,\delta\mathbf{x} = 0, \tag{83}$$

where again the Jacobian $J = (d\mathbf{h}/d\mathbf{x})$ appears.

We repeat the above process as λ is varied from 0 to 1 in small steps. The step size $\delta\lambda$ can be adjusted to keep the departure from the desired path smaller than some chosen threshold.

Perhaps the most awkward practical problem of continuation approach is "jumping" of a solution being traced from its correct path to a path that passes close to it. Path jumping can be detected when two paths end at the same solution and when that solution can be shown not to be a multiple root of the system of equations. Path jumping can also sometimes be detected by tracking solutions in reverse (that is as λ is decreased towards zero), and noting whether one returns to the starting solution. Something has gone awry when this does not happen. The probability of path jumping can be reduced by taking smaller steps, but this, of course, slows the computation.