International Journal of Computer Vision, 4, 59-78 (1990)

Relative Orientation*

BERTHOLD K.P. HORN
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139

Abstract
Before corresponding points in images taken with two cameras can be used to recover distances to objects in a scene, one has to determine the position and orientation of one camera relative to the other. This is the classic photogrammetric problem of *relative orientation*, central to the interpretation of binocular stereo information. Iterative methods for determining relative orientation were developed long ago; without them we would not have most of the topographic maps we do today. Relative orientation is also of importance in the recovery of motion and shape from an image sequence when successive frames are widely separated in time. Workers in motion vision are rediscovering some of the methods of photogrammetry.

Described here is a simple iterative scheme for recovering relative orientation that, unlike existing methods, does not require a good initial guess for the baseline and the rotation. The data required is a pair of bundles of corresponding rays from the two projection centers to points in the scene. It is well known that at least five pairs of rays are needed. Less appears to be known about the existence of multiple solutions and their interpretation. These issues are discussed here. The unambiguous determination of all of the parameters of relative orientation is not possible when the observed points lie on a *critical surface*. These surfaces and their degenerate forms are analyzed as well.

1 Introduction

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 the 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.

It is also the method used in motion vision when feature points are tracked and the image displacements that occur in the time between two successive frames are relatively large (see for example [53] and [52]). The connection between these two problems has not attracted much attention before, nor has the relationship of motion vision to some aspects of photogrammetry (but see [27]). It turns out, for example, that the well-known motion field equations [30, 4] are just the *parallax equations* of photogrammetry [14, 32] that occur in the incremental adjustment of relative orientation. Most papers on relative orientation give only the equation for y-parallax, corresponding to the equation for the y-component of the motion field (see for example the first equation in Gill [13], equation (1) in Jochmann [25], and equation (6) in Oswal [36]). Some papers actually give equations for both x- and y-parallax (see for example equation (9) in Bender [1]).

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 [12, 19, 32, 43, 44, 45, 49, 55].

Let us use the terms *left* and *right* to identify the two cameras (in the case of the application to motion vision

*This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0100, in part by the System Development Foundation, and in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number N00014-85-K-0124.

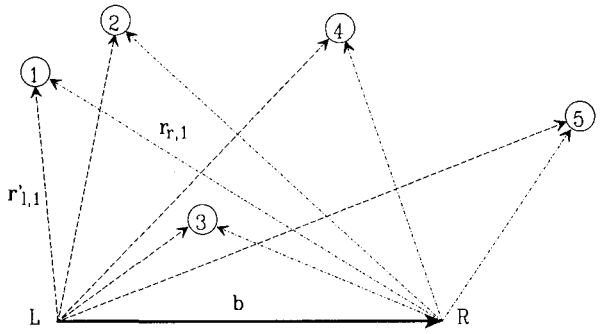

Fig. 1. Points in the environment are viewed from two camera positions. The relative orientation is the direction of the baseline **b**, and the rotation relating the left and right coordinate systems. The directions of rays to at least five scene points must be known in both camera coordinate systems.

these will be the camera positions and orientations corresponding to the earlier and the later frames respectively.[1] The ray from the center of projection of the left camera to the center of projection of the right camera is called the *baseline* (see figure 1). 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).[2] 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.[3]

2 Existing Solution Method

Various empirical procedures have been devised for determining the relative orientation in an analog fashion. Most commonly used are stereoplotters, optical devices that permit viewing of image pairs and superimposed synthetic features called floating marks. Differences in ray direction parallel to the baseline are called *horizontal disparities* (or x-parallaxes), while differences in ray direction orthogonal to the baseline are called *vertical disparities* (or y-parallaxes).[4] Horizontal disparities encode distances to points on the surface and are the quantities sought after in measurement of the underlying topography. There should be no vertical disparities when the device is adjusted to the correct relative orientation, since rays from the left and right projection center must lie in a plane that contains the baseline (an *epipolar plane*) if they are to intersect.

The methods used in practice to determine the correct relative orientation depend on successive adjustments to eliminate the vertical disparity at each of five or six image points that are arranged in one or another specially designed pattern [32, 39, 45, 50, 55]. In each of these adjustments, a single parameter of the relative orientation is varied in order to remove the vertical disparity at one of the points. Which adjustment is made to eliminate the vertical disparity at a specific point depends on the particular method chosen. In each case, however, one of the adjustments, rather than being guided visually, is made by an amount that is calculated, using the measured values of earlier adjustments. The calculation is based on the assumptions that the surface being viewed can be approximated by a plane, that the baseline is roughly parallel to this plane, and that the optical axes of the two cameras are roughly perpendicular to this plane.[5]

The whole process is iterative in nature, since the reduction of vertical disparity at one point by means of an adjustment of a single parameter of the relative orientation disturbs the vertical disparity at the other points. Convergence is usually rapid if a good initial guess is available. It can be slow, however, when the assumptions on which the calculation is based are violated, such as in "accidented" or hilly terrain [54]. These methods typically use Euler angles to represent three-dimensional rotations [26] (traditionally denoted by the Greek letters κ, ϕ, and ω). Euler angles have a

[1] In what follows we use the coordinate system of the right (or later) camera as the reference. One can simply interchange left and right if it happens to be more convenient to use the coordinate system of the left (or earlier) camera. The solution obtained in this fashion will be the exact inverse of the solution obtained the other way.

[2] Actually, any coordinate system rigidly attached to the image-forming system may be used.

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

[4] This naming convention stems from the observation that, in the usual viewing arrangement, horizontal disparities correspond to left–right displacements in the image, whereas vertical disparities correspond to up–down displacements.

[5] While these very restrictive assumptions are reasonable in the case of typical aerial photography, they are generally not reasonable in the case of terrestrial or industrial photogrammetry, or in robotics.

number of shortcomings for describing rotations that become particularly noticeable when these angles become large.[6]

There also exist related digital procedures that converge rapidly when a good initial guess of the relative orientation is available, as is usually the case when one is interpreting aerial photography [45]. Not all of these methods use Euler angles. Thompson [49], for example, uses twice the Gibb's vector [26] to represent rotations. These procedures may fail to converge to the correct solution when the initial guess is far off the mark. In the application to motion vision, approximate translational and rotational components of the motion are often not known initially, so a procedure that depends on good initial guesses is not particularly useful. Also, in terrestrial, close-range [35] and industrial photogrammetry [8] good initial guesses are typically harder to come by than they are in aerial photography.

Normally, the directions of the rays are obtained from points generated by projection onto a planar imaging surface. In this case the directions are confined to the field of view as determined by the active area of the image plane and its distance to the center of projection. The field of view is always less than a hemisphere, since only points in front of the camera can be imaged.[7] The method described here applies, however, no matter how the directions to points in the scene are determined. There is no restriction on the possible ray directions. We do assume, however, that we can tell which of two opposite semi-infinite line segments the point lies on. If a point lies on the correct line segment, we will say that it lies in *front* of the camera, otherwise it will be considered to be *behind* the camera (even when these terms do not strictly apply).

The problem of relative orientation is generally considered solved, and so has received little attention in the photogrammetric literature in recent times [54]. In the annual index of *Photogrammetric Engineering*, for example, there is only one reference to the subject in the last ten years [10] and six in the decade before that. This is very little in comparison to the large number of papers on this subject in the fifties, as well as the sixties, including those by Gill [13], Sailor [39], Jochmann [25], Ghosh [11], Forrest [7], and Oswal [36].

[6]The angles tend to be small in traditional applications to photographs taken from the air, but often are quite large in the case of terrestrial photogrammetry.

[7]The field of view is, however, larger than a hemisphere in some fish-eye lenses, where there is significant radial distortion.

In this paper we discuss the relationship of relative orientation to the problem of motion vision in the situation where the motion between the exposure of successive frames is relatively large. Also, a new iterative algorithm is described, as well as a way of dealing with the situation when there is no initial guess available for the rotation or the direction of the baseline. The advantages of the unit quaternion notation for representing rotations are illustrated as well. Finally, we discuss critical surfaces, surface shapes that lead to difficulties in establishing a unique relative orientation.

(One of the reviewers pointed out that L. Hinsken [16, 17] recently obtained a method for computing the relative orientation based on a parameterization of the rotation matrix that is similar to the unit quaternion representation used here. In his work, the unknown parameters are the rotations of the left and right cameras with respect to a coordinate system fixed to the baseline, while here the unknowns are the direction of the baseline and the rotation of a coordinate system fixed to one of the cameras in a coordinate system fixed to the other camera. Hinsken also addresses the simultaneous orientation of more than two bundles of rays, but says little about multiple solutions, critical surfaces, and methods for searching the space of unknown parameters.)

3 Coplanarity Condition

If the ray from the left camera and the corresponding ray from the right camera are to intersect, they must lie in a plane that also contains the baseline. Thus, if $\mathbf{b}$ is the vector representing the baseline, $\mathbf{r}_r$ is the ray from the right projection center to the point in the scene and $\mathbf{r}_l$ is the ray from the left projection center to the point in the scene, then the triple product

$$[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] \tag{1}$$

equals zero, where $\mathbf{r}_l' = \text{rot}(\mathbf{r}_l)$ is the left ray rotated into the right camera's coordinate system.[8] This is the *coplanarity condition* (see figure 2).

We obtain one such constraint from each pair of rays. There will be an infinite number of solutions for the baseline and the rotation when there are fewer than five pairs of rays, since there are five unknowns and each pair of rays yields only one constraint. Conversely, if

[8]The baseline vector $\mathbf{b}$ is here also assumed to be measured in the coordinate system of the right camera.

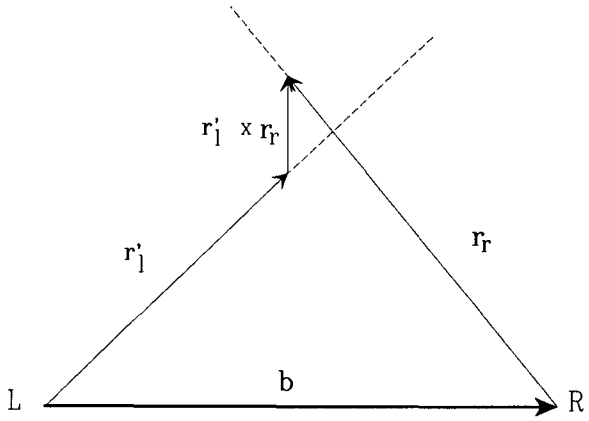

Fig. 2. Two rays approach closest where they are intersected by a line perpendicular to both. If there is no measurement error, and the relative orientation has been recovered correctly, then the two rays actually intersect. In this case the two rays and the baseline in a common plane.

there are more than five pairs of rays, the constraints are likely to be inconsistent as the result of small errors in the measurements. In this case, no exact solution of the set of constraint equations will exist, and it makes sense instead to minimize the sum of squares of errors in the constraint equations. In practice, one should use more than five pairs of rays in order to reduce the influence of measurement errors [25]. We shall see later that the added information also allows one to eliminate spurious apparent solutions.

In the above we have singled out one of the two image-forming systems to provide the reference coordinate system. It should be emphasized that we obtain the exact inverse of the solution if we chose to use a coordinate system aligned with the other image-forming system instead.

4 What Is the Appropriate Error Term?

In this section we discuss the weights w by which the squares of the triple products $[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]$ should be multiplied in the total sum of errors. The reader may wish to skip this section upon first reading, but keep in mind that there is some rational basis for choosing these weights. Also note that one can typically compute a good approximation to the exact least-squares solution without introducing the weighting factors.

The triple product $t = [\mathbf{b}\ \mathbf{r}_l'\mathbf{r}_r]$ is zero when the left and right ray are coplanar with the baseline. The triple product itself is, however, not the ideal measure of departure from best fit. It is worthwhile exploring the geometry of the two rays more carefully to see this. Consider the points on the rays where they approach each other the closest (see figure 2). The line connecting these points will be perpendicular to both rays, and hence parallel to $(\mathbf{r}_l' \times \mathbf{r}_r)$. As a consequence, we can write

$$\alpha\mathbf{r}_l' + \gamma(\mathbf{r}_l' \times \mathbf{r}_r) = \mathbf{b} + \beta\mathbf{r}_r \qquad (2)$$

where α and β are proportional to the distances along the left and the right ray to the points where they approach most closely, while γ is proportional to the shortest distance between the rays. We can find γ by taking the dot-product of the equality above with $\mathbf{r}_l' \times \mathbf{r}_r$. We obtain

$$\gamma\,\|\mathbf{r}_l' \times \mathbf{r}_r\|^2 = [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] \qquad (3)$$

Similarly, taking dot-products with $\mathbf{r}_r \times (\mathbf{r}_l' \times \mathbf{r}_r)$ and $\mathbf{r}_l' \times (\mathbf{r}_l' \times \mathbf{r}_r)$, we obtain

$$\begin{aligned}\alpha\,\|\mathbf{r}_l' \times \mathbf{r}_r\|^2 &= (\mathbf{b} \times \mathbf{r}_r) \cdot (\mathbf{r}_l' \times \mathbf{r}_r) \\ \beta\,\|\mathbf{r}_l' \times \mathbf{r}_r\|^2 &= (\mathbf{b} \times \mathbf{r}_l') \cdot (\mathbf{r}_l' \times \mathbf{r}_r)\end{aligned} \qquad (4)$$

Clearly, $\alpha\,\|\mathbf{r}_l'\|$ and $\beta\,\|\mathbf{r}_r\|$ are the distances along the rays to the points of closest approach.[9]

Later we will be more concerned with the signs of α and β. Normally, the points where the two rays approach the closest will be in front of both cameras, that is, both α and β will be positive. If the estimated baseline or rotation is in error, however, then it is possible for one or both of the calculated parameters α and β to come out negative. We will use this observation later to distinguish among different apparent solutions. We will call a solution where all distances are positive a *positive solution*. In photogrammetry one is typically only interested in positive solutions.

The perpendicular distance between the left and the right ray is

$$d = \gamma\,\|\mathbf{r}_l' \times \mathbf{r}_r\| = \frac{[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]}{\|\mathbf{r}_l' \times \mathbf{r}_r\|} \qquad (5)$$

This distance itself, however, is also not the ideal measure of departure from best fit, since the measurement errors are in the image, not in the scene (see also the discussion in [9]. A least-squares procedure should be based on the error in determining the direction of the rays, not on the distance of closest approach. We need

[9]The dot-products of the cross-products can, of course, be expanded out in terms of differences of products of dot-products.

to relate variations in ray direction to variations in the perpendicular distance between the rays, and hence the triple product.

Suppose that there is a change $\delta\theta_l$ in the vertical disparity of the left ray direction and $\delta\theta_r$ in the vertical disparity of the right ray direction. That is, $\mathbf{r}_l'$ and $\mathbf{r}_r$ are changed by adding

$$\delta\mathbf{r}_l' = \frac{\mathbf{r}_l' \times \mathbf{r}_r}{\|\mathbf{r}_l' \times \mathbf{r}_r\|} \|\mathbf{r}_l'\| \, \delta\theta_l$$
$$\delta\mathbf{r}_r = \frac{\mathbf{r}_l' \times \mathbf{r}_r}{\|\mathbf{r}_l' \times \mathbf{r}_r\|} \|\mathbf{r}_r\| \, \delta\theta_r \tag{6}$$

respectively. Then, from figure 3, we see that the change in the perpendicular distance d is just $\delta\theta_l$ times the distance form the left center of projection to the point of closest approach on the left ray, minus $\delta\theta_r$ times the distance from the right center of projection to the point of closest approach on the right ray, or

$$\delta\mathrm{d} = \alpha \|\mathbf{r}_l'\| \, \delta\theta_l - \beta \|\mathbf{r}_r\| \, \delta\theta_r \tag{7}$$

From equation (5) we see that the corresponding change in the triple product is

$$\delta t = \|\mathbf{r}_l' \times \mathbf{r}_r\| \, \delta d$$

Thus if the variance in the determination of the vertical disparity of the left ray is σ_l^2, and the variance in the determination of the vertical disparity of the right ray is σ_r^2, then the variance in the triple product will be[10]

$$\sigma_t^2 = \|\mathbf{r}_l' \times \mathbf{r}_r\|^2 \, \sigma_d^2 \tag{8}$$

or

$$\sigma_t^2 = \|\mathbf{r}_l' \times \mathbf{r}_r\|^2 \, (\alpha^2 \|\mathbf{r}_l'\|^2 \, \sigma_l^2 + \beta^2 \|r_r\|^2 \, \sigma_r^2) \tag{9}$$

This implies that we should apply a weight

$$w = \sigma_0^2 / \sigma_t^2 \tag{10}$$

to the square of each triple product in the sum to be minimized, where σ_0^2 is arbitrary (see Mikhail and Ackerman [31], p. 65). Written out in full we have

[10]The error in determining the direction of a ray depends on image position, since a fixed interval in the image corresponds to a larger angular interval in the middle of the image than it does at the periphery. The reason is that the middle of the image is closer to the center of projection than is the periphery. In any case, one can determine what the variance of the error in vertical disparity is, given the image position and the estimated error in determining positions in the image.

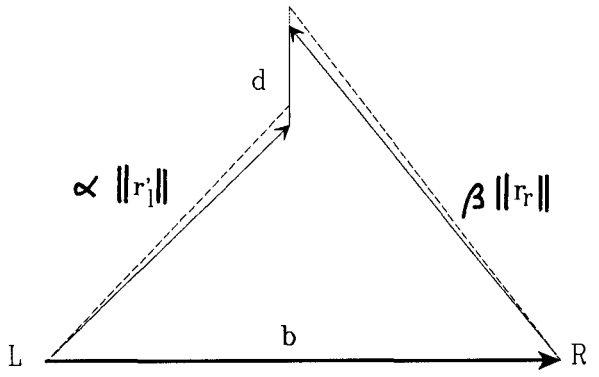

Fig. 3. Variations in the triple product $t = [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]$ can be related to variations in the perpendicular distance d between the two rays. Variations in this distance, in turn, can be related to variations in the measurement of the directions of the left and right rays. These relationships can be used to arrive at weighting factors that allow minimization of errors in image positions while working with the sums of squares of triple products.

$$w = \|\mathbf{r}_l' \times \mathbf{r}_r\|^2 \sigma_0^2 \div [((\mathbf{b} \times \mathbf{r}_r) \cdot (\mathbf{r}_l' \times \mathbf{r}_r))^2 \|\mathbf{r}_l'\|^2 \sigma_l^2 + ((\mathbf{b} \times \mathbf{r}_l') \cdot (\mathbf{r}_l' \times \mathbf{r}_r))^2 \|\mathbf{r}_r\|^2 \sigma_r^2] \tag{11}$$

(Note again that errors in the horizontal disparity do not influence the computed relative orientation; instead they influence errors in the distances recovered using the relative orientation).

Introduction of the weighting factors makes the sum to be minimized quite complicated, since changes in baseline and rotation affect both numerators and denominators of the terms in the total error sum. Near the correct solution, the triple products will be small and so changes in the estimated rotation and baseline will tend to induce changes in the triple products that are relatively large compared to the magnitudes of the triple products themselves. The changes in the weights, on the other hand, will generally be small compared to the weights themselves. This suggests that one should be able to treat the weights as constant during a particular iterative step.

Also note that one can compute a good approximation to the solution without introducing the weighting factors at all. This approximation can then be used to start an iterative procedure that does take the weights into account, but treats them as constant during each iterative step. This works well because changes in the weights become relatively small as the solution is approached.

5 Least Squares Solution for the Baseline

If the rotation is known, it is easy to find the best-fit baseline, as we show next. This is useful, despite the fact that we do not usually know the rotation. The reason is that the ability to find the best baseline, given a rotation, reduces the dimensionality of the search space from five to three. This makes it much easier to systematically explore the space of possible starting values for the iterative algorithm.

Let $\{\mathbf{r}_{l,i}\}$ and $\{\mathbf{r}_{r,i}\}$, for $i = 1, \ldots, n$, be corresponding bundles of left and right rays. We wish to minimize

$$E = \sum_{i=1}^{n} w_i[\mathbf{b}\ \mathbf{r}'_{l,i}\ \mathbf{r}_{r,i}]^2 = \sum_{i=1}^{n} w_i(\mathbf{b} \cdot (\mathbf{r}'_{l,i} \times \mathbf{r}_{r,i}))^2 \quad (12)$$

subject to the condition $\mathbf{b} \cdot \mathbf{b} = 1$, where $\mathbf{r}'_{l,i}$ is the rotated left ray $\mathbf{r}_{l,i}$ as before. If we let $\mathbf{c}_i = \mathbf{r}'_{l,i} \times \mathbf{r}_{r,i}$, we can rewrite the sum in the simpler form

$$E = \sum_{i=1}^{n} w_i(\mathbf{b} \cdot \mathbf{c}_i)^2 = \mathbf{b}^T \left(\sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{c}_i^T \right) \mathbf{b} \quad (13)$$

where we have used the equivalence $\mathbf{b} \cdot \mathbf{c}_i = \mathbf{b}^T\mathbf{c}_i$, which depends on the interpretation of column vectors as 3×1 matrixes. The term $\mathbf{c}_i\mathbf{c}_i^T$ is a dyadic product, a 3×3 matrix obtained by multiplying a 3×1 matrix by a 1×3 matrix.

The error sum is a quadratic form involving the real symmetric matrix.[11]

$$C = \sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{c}_i^T \quad (14)$$

The minimum of such a quadratic form is the smallest eigenvalue of the matrix C, attained when $\mathbf{b}$ is the corresponding unit eigenvector (see, for example, the discussion of Rayleigh's quotient in Korn and Korn [26]. This can be verified by introducing a Lagrangian multiplier λ and minimizing

$$E' = \mathbf{b}^T C \mathbf{b} + \lambda(1 - \mathbf{b}^T\mathbf{b}) \quad (15)$$

subject to the condition $\mathbf{b}^T\mathbf{b} = 1$. Differentiating with respect to $\mathbf{b}$ and setting the result equal to zero yields

$$C\mathbf{b} = \lambda\mathbf{b} \quad (16)$$

The error corresponding to a particular solution of this equation is found by premultiplying by $\mathbf{b}^T$:

$$E = \mathbf{b}^T C \mathbf{b} = \lambda \mathbf{b}^T\mathbf{b} = \lambda \quad (17)$$

The three eigenvalues of the real symmetric matrix C are non-negative, and can be found in closed form by solving a cubic equation, while each of the corresponding eigenvectors has components that are the solution of three homogeneous equations in three unknowns [26]. If the data are relatively free of measurement error, then the smallest eigenvalue will be much smaller than the other two, and a reasonable approximation to the sought-after result can be obtained by solving for the eigenvector using the assumption that the smallest eigenvalue is actually zero. This way one need not even solve the cubic equation (see also Horn and Weldon [24]).

If $\mathbf{b}$ is a unit eigenvector, so is $-\mathbf{b}$. Changing the sense of the baseline does not change the magnitude of the error term $[\mathbf{b}\ \mathbf{r}'_l\ \mathbf{r}_r]$. It does, however, change the signs of α, β, and γ. One can decide which sense of the baseline direction is appropriate by determining the signs of α_i and β_i for $i = 1, \ldots, n$. Ideally, they should all be positive, but when the baseline and the rotation are incorrect they may not be.

The solution for the optimal baseline is not unique unless there are at least two pairs of corresponding rays. The reason is that the eigenvector we are looking for is not uniquely determined if more than one of the eigenvalues is zero, and the matrix has rank less than two if it is the sum of fewer than two dyadic products of independent vectors. This is not a significant restriction, however, since we need at least five pairs of rays to solve for the rotation anyway.

6 Iterative Improvement of Relative Orientation

If one ignores the orthonormality of the rotation matrix, a set of nine homogeneous linear equations can be obtained by a transformation of the coplanarity conditions that was first described by Thompson [49]. These equations can be solved when eight pairs of corresponding ray directions are known [38, 27]. This is not a least-squares method that can make use of redundant measurements, nor can it be applied when fewer than eight points are given. Also, the method is strongly affected by measurement errors and fails for certain configurations of points [28].

[11] The terms in the sum of dyadic products forming the matrix C contain the weights discussed in the previous section. This only makes sense, however, if a guess is already available for the baseline—unit weights may be used otherwise.

No true closed-form solution of the least-squares problem has been found for the general case, where both baseline and rotation are unknown. However, it is possible to determine how the overall error is affected by small changes in the baseline and small changes in the rotation. This allows one to make iterative adjustments to the baseline and the rotation that reduce the sum of squares of errors.

We can represent a small change in the baseline by an infinitesimal quantity $\delta\mathbf{b}$. If this change is to leave the length of the baseline unaltered, then

$$\|\mathbf{b} + \delta\mathbf{b}\|^2 = \|\mathbf{b}\|^2 \tag{18}$$

or

$$\|\mathbf{b}\|^2 + 2\mathbf{b} \cdot \delta\mathbf{b} + \|\delta\mathbf{b}\|^2 = \|\mathbf{b}\|^2 \tag{19}$$

If we ignore quantities of second-order, we obtain

$$\delta\mathbf{b} \cdot \mathbf{b} = 0$$

that is, $\delta\mathbf{b}$ must be perpendicular to $\mathbf{b}$.

A small change in the rotation can be represented by an infinitesimal rotation vector $\delta\omega$. The direction of this vector is parallel to the axis of rotation, while its magnitude is the angle of rotation. This incremental rotation takes the rotated left ray, $\mathbf{r}_l'$, into

$$\mathbf{r}_l'' = \mathbf{r}_l' + (\delta\omega \times \mathbf{r}_l') \tag{20}$$

This follows from Rodrigues' formula for the rotation of a vector $\mathbf{r}$

$$\cos\theta\,\mathbf{r} + \sin\theta\,(\omega \times \mathbf{r}) + (1 - \cos\theta)(\omega \cdot \mathbf{r})\omega \tag{21}$$

if we let $\theta = \|\delta\omega\|$, $\omega = \delta\omega/\|\delta\omega\|$, and note that $\delta\omega$ is an infinitesimal quantity. Finally then, we see that $t = [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]$ becomes

$$t + \delta t = [(\mathbf{b} + \delta\mathbf{b})\ (\mathbf{r}_l' + \delta\omega \times \mathbf{r}_l')\ \mathbf{r}_r] \tag{22}$$

or,

$$[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] + [\delta\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] + [\mathbf{b}\ (\delta\omega \times \mathbf{r}_l')\ \mathbf{r}_r] \tag{23}$$

if we ignore the term $[\delta\mathbf{b}\ (\delta\omega \times \mathbf{r}_l')\ \mathbf{r}_r]$, because it contains the product of two infinitesimal quantities. We can expand two of the triple products in the expression above and obtain

$$[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] + (\mathbf{r}_l' \times \mathbf{r}_r) \cdot \delta\mathbf{b} + (\mathbf{r}_r \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{r}_l') \tag{24}$$

or

$$t + \mathbf{c} \cdot \delta\mathbf{b} + \mathbf{d} \cdot \delta\omega \tag{25}$$

for short, where

$$t = [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r],\ \mathbf{c} = \mathbf{r}_l' \times \mathbf{r}_r,\ \mathbf{d} = \mathbf{r}_l' \times (\mathbf{r}_r \times \mathbf{b}) \tag{26}$$

Now, we are trying to minimize

$$E = \sum_{i=1}^{n} w_i(t_i + \mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\omega)^2 \tag{27}$$

subject to the condition $\mathbf{b} \cdot \delta\mathbf{b} = 0$. We can introduce a Lagrange multiplier in order to deal with the constraint. Instead of minimizing E itself, we then have to minimize

$$E' = E + 2\lambda(\mathbf{b} \cdot \delta\mathbf{b}) \tag{28}$$

(where the factor of two is introduced to simplify the algebra later). Differentiating E' with respect to $\delta\mathbf{b}$, and setting the result equal to zero yields

$$\sum_{i=1}^{n} w_i(t_i + \mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\omega)\mathbf{c}_i + \lambda\mathbf{b} = \mathbf{0} \tag{29}$$

By taking the dot-product of this expression with $\mathbf{b}$, and using the fact that $\mathbf{b} \cdot \mathbf{b} = 1$ one can see that

$$\lambda = -\sum_{i=1}^{n} w_i(t_i + \mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\omega)t_i \tag{30}$$

which means that $-\lambda$ is equal to the total error when one is at a stationary point, where $\delta\mathbf{b}$ and $\delta\omega$ are equal to zero.

If we differentiate E' with respect to $\delta\omega$ and set this result also equal to zero, we obtain

$$\sum_{i=1}^{n} w_i(t_i + \mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\omega)\mathbf{d}_i = \mathbf{0} \tag{12}$$

Finally, if we differentiate E' with respect to λ we get back the constraint

$$\mathbf{b} \cdot \delta\mathbf{b} = 0 \tag{32}$$

The two vector equations and the one scalar equation (equations (29), (31), and (32)) constitute seven linear scalar equations in the six unknown components of $\delta\mathbf{b}$ and $\delta\omega$ and the unknown Lagrangian multiplier λ. We can rewrite them in the more compact form:

$$\begin{aligned} C\,\delta\mathbf{b} + F\,\delta\omega + \lambda\mathbf{b} &= -\bar{\mathbf{c}} \\ F^T\,\delta\mathbf{b} + D\,\delta\omega &= -\bar{\mathbf{d}} \\ \mathbf{b}^T\,\delta\mathbf{b} &= 0 \end{aligned} \tag{33}$$

or

$$\begin{bmatrix} C & F & \mathbf{b} \\ F^T & D & \mathbf{0} \\ \mathbf{b}^T & \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{b} \\ \delta\boldsymbol{\omega} \\ \lambda \end{bmatrix} = - \begin{bmatrix} \bar{\mathbf{c}} \\ \bar{\mathbf{d}} \\ 0 \end{bmatrix} \tag{34}$$

where

$$C = \sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{c}_i^{\mathrm{T}}$$

$$F = \sum_{i=1}^{n} w_i \mathbf{c}_i \mathbf{d}_i^T \tag{35}$$

$$D = \sum_{i=1}^{n} w_i \mathbf{d}_i \mathbf{d}_i^T$$

while

$$\bar{\mathbf{c}} = \sum_{i=1}^{n} w_i t_i \mathbf{c}_i \quad \text{and} \quad \bar{\mathbf{d}} = \sum_{i=1}^{n} w_i t_i \mathbf{d}_i \tag{36}$$

The above gives us a way of finding small changes in the baseline and rotation that reduce the overall error sum.[12] The equations shown (equation (34)) are the *symmetric normal equations* (see also Mikhail and Ackerman [31], p. 229) and yield incremental adjustments for the rotation and the baseline.[13] This method can be applied iteratively to locate a minimum. Numerical experiments confirm that it converges rapidly when a good initial guess is available.

7 Singularities and Sensitivity to Errors

The computation of the incremental adjustments cannot be carried out with precision when the coefficient matrix becomes ill conditioned. This occurs when there are fewer than five pairs of rays, as well as for certain rare configurations of points in the scene (see the discussion of *critical surfaces* later). The coefficient matrix may also become ill conditioned when the iterative process approaches a stationary point that is not a minimum, as is often found between two nearby local minima. At such points the total error will typically still be quite large, yet vary rather slowly over a significant region of parameter space. In this situation the correction terms $\delta\mathbf{b}$ and $\delta\boldsymbol{\omega}$ that are computed may become very large. Since the whole method is based on the assumption that these adjustments are small, it is important to limit their magnitude.[14]

To guard against bad data points (and local minima of the error function) it is important to compute the total error before accepting a solution. It should be compared against what is expected, given the variance of the error in the vertical disparity of the ray directions. The estimate $\hat{\sigma}_0^2$ of the variance factor σ_0^2 can be obtained from the weighted error sum

$$E = \sum_{i=1}^{n} w_i [\mathbf{b}\ \mathbf{r}'_{l,i}\ \mathbf{r}_{r,i}]^2 \tag{37}$$

using the updated values of the rotation and the baseline, or from the approximation

$$E = \sum_{i=1}^{n} w_i (t_i + \mathbf{c}_i \cdot \delta\mathbf{b} + \mathbf{d}_i \cdot \delta\boldsymbol{\omega})^2 \tag{38}$$

using the computed increments $\delta\mathbf{b}$ and $\delta\boldsymbol{\omega}$, and the old values of the rotation and the baseline. We have

$$\hat{\sigma}_0^2 = E/(n - 5) \tag{39}$$

where n is the number of pairs of rays (see also Mikhail and Ackerman [31], p. 115). A χ^2-test with five degrees of freedom can be applied to $\hat{\sigma}_0^2/\sigma_0^2$ to test whether the estimated variance factor deviates significantly from the assumed value.

The inverse of the normal matrix introduced above has great significance, since from it can be derived the covariance matrix for the unknown orientation parameters (the elements of $\delta\mathbf{b}$ and $\delta\boldsymbol{\omega}$) using the covariance matrix of the quantities ($\bar{\mathbf{c}}$ and $\bar{\mathbf{d}}$) appearing on the right-hand side of the normal equations (see also [31], p. 230). It is important to point out, however, that the variances of the six parameters do not tell the whole story,

[12]It is also possible to reduce the problem to the solution of six linear equations in six unknowns by first eliminating the Lagrangian multiplier λ using $\mathbf{b} \cdot \mathbf{b} = 1$ [22], but this leads to an asymmetrical coefficient matrix that requires more work to set up. One of the reviewers pointed out that the symmetric normal equations can be solved directly, as shown above.

[13]Note that the customary savings of about half the computation when solving a system of equations with symmetric coefficient matrix cannot be fully achieved here since the last element on the main diagonal is zero. It may also be of interest to note that the top left 6×6 submatrix has at most rank n, since it is the sum of n dyadic products—it thus happens to be singular when $n = 5$.

[14]The exact size of the limit is not very important, a limit between 1/10 and 1 on the combined magnitudes of $\delta\mathbf{b}$ and $\delta\boldsymbol{\omega}$ appears to work quite well.

since the covariances (off-diagonal elements) can become very large, particularly in ill-conditioned cases. In such cases, the total error may vary appreciably when any one of the parameters is changed individually, yet a carefully chosen combination of changes in the parameters may leave the total error almost unchanged. In these situations, movement along special directions in parameter space may yield changes in total error that are a million-fold smaller than changes induced by movement in other directions. This means that for the same change in the total error, movement in parameter space in these special directions can be a thousand-fold larger than in other directions.

Ideally, a sensitivity analysis should be performed to check the stability of the solution [6].

8 Adjusting the Baseline and the Rotation

The iterative adjustment of the baseline is straightforward:

$$\mathbf{b}^{n+1} = \mathbf{b}^n + \delta\mathbf{b}^n \tag{40}$$

where $\mathbf{b}^n$ is the baseline estimate at the beginning of the nth iteration, while $\delta\mathbf{b}^n$ is the adjustment computed during the nth iteration, as discussed in the previous section. If $\delta\mathbf{b}^n$ is not infinitesimal, the result will not be a unit vector. We can, and should, normalize the result by dividing by its magnitude.

8.1 Adjustment of Rotation Using Unit Quaternions

Adjusting the rotation is a little harder. Rotations are conveniently represented by unit quaternions [19, 20, 40, 46, 47]. The groundwork for the application of the unit quaternion notation in photogrammetry was laid by Thompson [48], Schut [42], and Pope [37]. A positive rotation about the axis ω through an angle θ is represented by the unit quaternion

$$\mathring{q} = \cos(\theta/2) + \sin(\theta/2)\omega \tag{41}$$

where ω is assumed to be a unit vector. Composition of rotations corresponds to multiplication of the corresponding unit quaternions. The rotated version of a vector $\mathbf{r}$ is computed using

$$\mathring{r}_n' = \mathring{q}\mathring{r}\mathring{q}^* \tag{42}$$

where $\mathring{q}^*$ is the conjugate of the quaternion $\mathring{q}$, that is, the quaternion obtained by changing the sign of the vector part. Here, $\mathring{r}$ is a purely imaginary quaternion with vector part $\mathbf{r}$, while $\mathring{r}'$ is a purely imaginary quaternion with vector part $\mathbf{r}'$. The above can also be written in the form

$$\mathbf{r}' = (q_0^2 - \mathbf{q}\cdot\mathbf{q})\mathbf{r} + 2(\mathbf{q}\cdot\mathbf{r})\mathbf{q} + 2q_0(\mathbf{q}\times\mathbf{r}) \tag{43}$$

where q_0 and $\mathbf{q}$ are the scalar and vector parts of the unit quaternion $\mathring{\mathbf{q}}$ (see also [19]).

The infinitesimal rotation $\delta\omega$ corresponds to the quarternion

$$\delta\mathring{\omega} = 1 + \frac{1}{2}\delta\omega \tag{44}$$

We can adjust the rotation $\mathring{q}$ by premultiplying with $\delta\mathring{\omega}$, that is,

$$\mathring{q}^{n+1} = \delta\mathring{\omega}^n\mathring{q}^n \tag{45}$$

If $\delta\omega^n$ is not infinitesimal, $\delta\mathring{\omega}^n$ will not be a unit quaternion, and so the result of the adjustment will not be a unit quaternion either. This undesirable state of affairs can be avoided by using either of the two unit quaternions

$$\delta\mathring{\omega} = \sqrt{1 - \frac{1}{4}\|\delta\omega\|^2} + \frac{1}{2}\delta\omega \tag{46}$$

or

$$\delta\mathring{\omega} = \left(1 + \frac{1}{2}\delta\omega\right)\Big/\sqrt{1 + \frac{1}{4}\|\delta\omega\|^2} \tag{47}$$

Alternatively, one can simply normalize the product by dividing by its magnitude.

8.2 Adjustment of Rotation Using Orthonormal Matrixes

The adjustment of rotation is a little trickier if orthonormal matrixes are used to represent rotations. We can write the relationship

$$\mathbf{r}' = \mathbf{r} + (\delta\omega \times \mathbf{r}) \tag{48}$$

in the form

$$\mathbf{r}' = \mathbf{r} + W\mathbf{r} \tag{49}$$

where the skew-symmetric matrix W is defined by

$$W = \begin{bmatrix} 0 & -\delta\omega_z & \delta\omega_y \\ \delta\omega_z & 0 & -\delta\omega_x \\ -\delta\omega_y & \delta\omega_x & 0 \end{bmatrix} \tag{50}$$

in terms of the components of rotation vector $\delta\omega = (\delta\omega_x, \delta\omega_y, \delta\omega_z)^T$. Consequently we may write $\mathbf{r}' = Q\mathbf{r}$, where $Q = I + W$, or

$$Q = \begin{bmatrix} 1 & -\delta\omega_z & \delta\omega_y \\ \delta\omega_z & 1 & -\delta\omega_x \\ -\delta\omega_y & \delta\omega_x & 1 \end{bmatrix} \tag{51}$$

One could then attempt to adjust the rotation by multiplication of the matrixes Q and R as follows:

$$R^{n+1} = Q^n R^n \tag{52}$$

The problem is that Q is not orthonormal unless $\delta\omega$ is infinitesimal. In practice this means that the rotation matrix will depart more and more from orthonormality as more and more iterative adjustments are made. It is possible to renormalize this matrix by finding the nearest orthonormal matrix, but this is complicated, since it involves the determination of the square-root of a symmetric matrix [23].[15]

To avoid this problem, we should really start with an orthonormal matrix to represent the incremental rotation. We can use either of the two unit quaternions in equations (46) or (47) to construct the corresponding orthonormal matrix

$$Q = \begin{bmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_y q_x + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_z q_x - q_0 q_y) & 2(q_z q_y + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{53}$$

where q_0 is the scalar part of the quaternion $\delta\mathring{\omega}$, while q_x, q_y, q_z are the components of the vector part.[16] Then the adjustment of rotation is accomplished using

$$R^{n+1} = Q^n R^n \tag{54}$$

Note, however, that the resulting matrixes will still tend to depart slightly from orthonormality due to numerical inaccuracies. This may be a problem if many iterations are required.

9 Ambiguities

9.1 Inherent Ambiguities and Dual Solution

The iterative adjustment described above may arrive at a number of apparently different solutions. Some of these are just different representations of the same solution, while others are related to the correct solution by a simple transformation. First of all, note that $-\mathring{q}$ represents the same rotation as $\mathring{q}$, since

$$(-\mathring{q})\ \mathring{r}\ (-\mathring{q}^*) = \mathring{q}\mathring{r}\mathring{q}^* \tag{55}$$

That is, antipodal points on the unit sphere in four dimensions represent the same rotation. If desired, one can prevent any confusion by ensuring that the first non-zero component of the resulting unit quaternion is positive, or that the largest component is positive.

Next, note that the triple product, $[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]$, changes sign, but not magnitude, when we replace $\mathbf{b}$ with $-\mathbf{b}$. Thus the two possible senses of the baseline yield the same sum of squares of errors. However, changing the sign of $\mathbf{b}$ does change the signs of both α and β. All scene points imaged are in front of the camera, so the distances should all be positive. In the presence of noise, it is possible that some of the distances turn out to be negative, but with reasonable data almost all of them should be positive. This normally allows one to pick the correct sense for the baseline.

Not so obvious is another possibility. Suppose we turn all of the left measurements through π radians about the baseline, in addition to the rotation already determined. That is, replace $\mathring{q}$ by $\mathring{q}' = \mathring{b}\mathring{q}$, where $\mathring{b}$ is a purely imaginary quaternion with vector part $\mathbf{b}$. The triple product can be written in the form

$$\begin{aligned} t = [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] &= \text{rot}\ (\mathbf{r}_l) \cdot (\mathbf{r}_r \times \mathbf{b}) \\ &= (\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot (\mathring{r}_r\mathring{b}) \end{aligned} \tag{56}$$

where $\mathring{r}_l$ and $\mathring{r}_r$ are purely imaginary quaternion with vector part $\mathbf{r}_l$ and $\mathbf{r}_r$ respectively. If we replace $\mathring{q}$ by $\mathring{q}' = \mathring{b}\mathring{q}$, we obtain for the triple product

$$t' = (\mathring{b}\mathring{q}\mathring{r}_l\mathring{q}^*\mathring{b}^*) \cdot (\mathring{r}_r\mathring{b}) = (\mathring{b}\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot (\mathring{r}_r\mathring{b}\mathring{b}) \tag{57}$$

or

$$t' = (\mathring{b}\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot (-\mathbf{b} \cdot \mathbf{b})\mathring{r}_r = -(\mathring{b}\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot \mathring{r}_r \tag{58}$$

or

$$t' = -(\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot (\mathring{b}^*\mathring{r}_r) = -(\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot (\mathring{r}_r\mathring{b}) = -t \tag{59}$$

where we have repeatedly used special properties of purely imaginary quaternions, as well as the fact that $\mathbf{b} \cdot \mathbf{b} = 1$. We conclude that the sign of the triple product is changed by the added rotation, but its magnitude is not. Thus the total error is undisturbed when the left rays are rotated through π radians about the baseline.

[15] This is another place where the unit quaternion representation has a distinct advantage: it is trivial to find the nearest unit quaternion to a quaternion that does not have unit magnitude.

[16] This expression for the orthonormal normal matrix in terms of the components of the corresponding unit quaternion can be obtained directly by expanding $\mathring{r}' = \mathring{q}\mathring{r}\mathring{q}^*$ or by means of Rodrigues' formula [19, 20].

The solution obtained this way will be called the *dual* of the other solution.

We can obtain the same result using vector notation: We replace $\mathbf{r}_l'$ with

$$\mathbf{r}_l'' = 2(\mathbf{b} \cdot \mathbf{r}_l')\mathbf{b} - \mathbf{r}_l' \tag{60}$$

using Rodrigues' formula for the rotation of a vector $\mathbf{r}$

$$\cos\theta\,\mathbf{r} + \sin\theta\,(\omega \times \mathbf{r}) + (1 - \cos\theta)(\omega \cdot \mathbf{r})\omega \tag{61}$$

with $\theta = \pi$ and $\omega = \mathbf{b}$. Then the triple product $[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]$ turns into

$$2(\mathbf{b} \cdot \mathbf{r}_l')[\mathbf{b}\ \mathbf{b}\ \mathbf{r}_r] - [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] = -[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] \tag{62}$$

This, once again, reverses the sign of the error term, but not its magnitude. Thus the sum of squares of errors is unaltered. The signs of α and β are affected, however, although this time not in as simple a way as when the sense of the baseline was reversed.

If $[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] = 0$, we find that exactly one of α and β changes sign. This can be shown as follows: The triple product will be zero when the left and right rays are coplanar with the baseline. In this case we have $\gamma = 0$, and so

$$\alpha\mathbf{r}_l' = \mathbf{b} + \beta\mathbf{r}_r \tag{63}$$

Taking the cross-product with $\mathbf{b}$ we obtain

$$\alpha(\mathbf{r}_l' \times \mathbf{b}) = \beta(\mathbf{r}_r \times \mathbf{b}) \tag{64}$$

If we now replace $\mathbf{r}_l'$ by $\mathbf{r}_l'' = 2(\mathbf{b} \cdot \mathbf{r}_l')\mathbf{b} - \mathbf{r}_l'$, we have for the new distances α' and β' along the rays:

$$-\alpha'(\mathbf{r}_l' \times \mathbf{b}) = \beta'(\mathbf{r}_r \times \mathbf{b}) \tag{65}$$

We conclude that the product $\alpha'\beta'$ has sign opposite to that of the product $\alpha\beta$. So if α and β are both positive, one of α' or β' must be negative.

In the presence of measurement error the triple product will not be exactly equal to zero. If the rays are nearly coplanar with the baseline, however, we find that one of α and β almost always changes sign. With very poor data, it is possible that both change sign.[17] In any case, we can reject a solution in which roughly half the distances are negative. Moreover, we can find the correct solution directly by introducing an additional rotation of π radians about the baseline, that is, by computing the dual of the solution.

[17]Even with totally random ray directions, however, this only happens 27.3% of the time, as determined by Monte Carlo simulation.

9.2 Remaining Ambiguity

If we take care of the three apparent two-way ambiguities discussed in the previous section, we find that in practice a unique solution is found, provided that a sufficiently large number of ray pairs are available. That is, the method converges to the unique global minimum from every possible starting point in parameter space.[18]

Several local minima in the sum of squares of errors appear when only a few more than the minimum of five ray pairs are available (as is common in practice). This means that one has to repeat the iteration with different starting values for the rotation in order to locate the global minimum. A starting value for the baseline can be found in each case using the closed-form method described in section 5. To search the parameter space effectively, one needs a way of efficiently sampling the space of rotations. The space of rotations is isomorphic to the unit sphere in four dimensions, with antipodal points identified. The rotation groups of the regular polyhedra provide convenient means of uniformly sampling the space of rotations. The group of rotations of the tetrahedron has 12 elements, that of the hexahedron and the octahedron has 24, and that of the icosahedron and the dodecahedron has 60 (representations of these groups are given in appendix A for convenience). One can use these as starting values for the rotation. Alternatively, one can just generate a number of randomly placed points on the unit sphere in four dimensions as starting values for the rotation.[19]

9.3 Number of Solutions Given Five Pairs of Rays

When there are exactly five pairs of rays, the situation is different again. In this case, we have five nonlinear equations (equation (1)) in five unknowns and so in general expect to find a finite number of exact solutions. That is, it is possible to find baselines and rotations that satisfy the coplanarity conditions *exactly* and reduce the sum of squares of errors to zero.

We can in fact easily express the coplanarity constraint as a polynomial in the components of $\mathbf{b}$ and $\mathring{q}$. Noting

[18]It has been shown that at most three essentially different relative orientations are compatible with a given sufficiently large number of ray pairs [29]—in practice one typically finds just one.

[19]We see here another advantage of the unit quaternion representation. It is not clear how one would sample the space of rotations using orthonormal matrixes directly.

that $\mathbf{r}_l' = \text{rot}\ (\mathbf{r}_l)$ and that the triple product can be written in the form

$$t = [\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] = (\mathring{q}\mathring{r}_l\mathring{q}^*) \cdot (\mathring{r}_r\mathring{b}) \tag{66}$$

we can expand equation (1), using equation (53), into

$$\begin{aligned}
&[(q_0^2 + q_x^2 - q_y^2 - q_z^2)l_x + 2(q_xq_y - q_0q_z)l_y \\
&+ 2(q_xq_z + q_0q_y)l_z \\
&\times (r_yb_z - r_zb_y) \\
&+ (2(q_yq_x + q_0q_z)l_z + (q_0^2 - q_x^2 + q_y^2 - q_z^2)l_y \\
&+ 2(q_yq_z - q_0q_x)l_z \\
&\times (r_zb_x - r_xb_z) \\
&+ [2(q_zq_x - q_0q_y)l_x + 2(q_zq_y + q_0q_x)l_y \\
&+ (q_0^2 - q_x^2 - q_y^2 + q_z^2)l_z \\
&\times (r_xb_y - r_yb_x) = 0
\end{aligned} \tag{67}$$

where $\mathbf{b} = (b_x, b_y, b_z)^T$, $\mathbf{r}_l = (l_x, l_y, l_z)^T$, $\mathbf{r}_r = (r_x, r_y, r_z)^T$, while $\mathring{q} = (q_0, q_x, q_y, q_z)^T$. This equation is linear in the components of $\mathbf{b}$ and quadratic in the components of $\mathring{q}$. When there are five ray pairs, there are five such equations. Together with the quadratic equations $\mathbf{b} \cdot \mathbf{b} = 1$ and $\mathring{q} \cdot \mathring{q} = 1$, they constitute seven polynomial equations in the seven components of $\mathbf{b}$ and $\mathring{q}$. An upper bound on the number of solutions is given by the product of the orders of the equations, which is $2^7 = 128$ in this case.[20] Note however that the equations are not changed if we change the sign of either $\mathbf{b}$ or $\mathring{q}$. Taking this into account, we see that there can be at most 32 distinct solutions. Not all of these need be real, of course.

In practice it is found that the number of solutions is typically a multiple of four (if we ignore reversals of $\mathring{q}$ and $\mathbf{b}$). With randomly chosen ray directions, about half of the cases lead to eight solutions, slightly more than a quarter have four solutions, while slightly less than a quarter have twelve. Less frequent are cases with sixteen solutions and a very small number of randomly generated test cases lead to twenty solutions, which appears to be the maximum number possible. Ray bundles for which there are *no* solutions at all are equally rare, but do exist. When one of the solutions corresponds to a critical surface, then the number of solutions is not a multiple of four—such cases correspond to places in ray parameter space that lie on the border between regions in which the number of solutions are different multiples of four. In this situation, small changes in the ray directions increase or decrease the number of solutions by two.

It has been brought to my attention, after receiving the comments of the reviewers, that it has recently been shown [5] that there can be at most twenty solutions of the relative orientation problem when $n = 5$, and that there exist pairs of ray bundles that actually lead to twenty solutions [34].[21] Typically there is one positive solution (or none), although several positive solutions may exist for a given set of ray bundles.

The ambiguities discussed above are, of course, of little concern if a reasonable initial guess is available. Note that methods that apply to the special case when there are five pairs of rays do not generalize to the least-squares problem when a larger number of ray pairs are available. In practice one should use more than five ray pairs, both to improve accuracy and to have a way of judging how large the errors might be.

10 Summary of the Algorithm

Consider first the case where we have an initial guess for the rotation. We start by finding the best-fit baseline direction using the closed-form method described in section 5. We may wish to determine the correct sense of the baseline by choosing the one that makes most of the signs of the distances positive. Then we proceed as follows:

- For each pair of corresponding rays, we compute $\mathbf{r}_{l,i}'$, the left ray direction $\mathbf{r}_{l,i}$ rotated into the right camera coordinate system, using the present guess for the rotation (equations (42), (43) or using equation (53)).
- We then compute the cross-product $\mathbf{c}_i = \mathbf{r}_{l,i}' \times \mathbf{r}_{r,i}$, the double cross-product $\mathbf{d}_i = \mathbf{r}_{l,i}' \times (\mathbf{r}_{r,i} \times \mathbf{b})$ and the triple-product $t_i = [\mathbf{b}\ \mathbf{r}_{l,i}'\ \mathbf{r}_{r,i}]$.
- If desired, we then compute the appropriate weighting factor w_i as discussed in section 4 (equation (11)).
- We accumulate the (weighted) dyadic products $w_i\mathbf{c}_i\mathbf{c}_i^T$, $w_i\mathbf{c}_i\mathbf{d}_i^T$ and $w_i\mathbf{d}_i\mathbf{d}_i^T$, as well as the (weighted) vectors $w_it_i\mathbf{c}_i$ and $w_it_i\mathbf{d}_i$. The totals of these quantities over all ray pairs give us the matrixes C, F, D and the vectors $\bar{\mathbf{c}}$ and $\bar{\mathbf{d}}$ (equations (35) and (36)).

[20]The three components of the baseline vector $\mathbf{b}$ can be eliminated fairly easily, because the equations are linear and homogeneous in these components. This leaves a smaller number of higher-order equations in the four components of $\mathring{q}$.

[21]Since dual solutions (obtained by rotating the left ray bundle through π about the baseline) are apparently not counted by these authors, they actually claim that the maximum number of solutions is ten.

- We can now solve for the increment in the baseline $\delta\mathbf{b}$ and the increment in the rotation $\delta\boldsymbol{\omega}$ using the method derived in section 6 (equation (34)).
- We adjust the baseline and the rotation using the methods discussed in section 8 (equations (40), (45) or (53), (54)), and recompute the sum of the squares of the error terms (equation (12)).

The new orientation parameters are then used in the next iteration of the above sequence of steps. As is the case with many iterative procedures, it is important to know when to stop. One could stop after either a fixed number of iterations or when the error becomes less than some predetermined threshold. Another approach would be to check on the size of the increments in the baseline and the rotation. These become smaller and smaller as the solution is approached, although their absolute size does not appear to provide a reliable stopping criterion.

The total error typically becomes small after a few iterations and no longer decreases at each step, because of limited accuracy in the arithmetic operations. So one could stop the iteration the first time the error increases. The total error may, however, also increase when the surfaces of constant error in parameter space are very elongated in certain directions, as happens when the problem is ill conditioned. In this case a step in the direction of the local gradient can cause one to skip right across the local "valley floor." It is thus wise to first check whether smaller steps in the given direction reduce the total error. The iteration is only stopped when small steps also increase the error.

When the decision has been made to stop the iteration, a check of the signs of the distances along the rays is in order. If most of them are negative, the baseline direction should be reversed. If neither sense of the baseline direction yields mostly positive distances, one needs to consider the dual solution (rotation of the left ray bundle through π radians about the baseline $\mathbf{b}$).

It makes sense also to check whether the solution is reasonable or whether it has perhaps been spoiled by some gross error in the data, such as incorrect correspondence between rays. When more than five pairs of rays are available, recomputation of the result using subsets obtained by omitting one ray at a time yield useful test results. These computations do not take much work, since a good guess for the solution is available in each case.

It is, of course, also useful to compute the total error E and to estimate the variance factor, as suggested in section 7. Finally, it may be desirable to estimate the standard deviations of the error in the six unknown parameters using the inverse of the matrix of coefficients of the symmetric normal equations, as indicated in section 7.

11 Search of Parameter Space and Statistics

If an initial guess is not available, one proceeds as follows:

- For each rotation in the chosen group of rotations, perform the above iteration to obtain a candidate baseline and rotation.
- Choose the solution that has all positive signs of the distances along rays and yields the smallest total error.

When there are many pairs of rays, the iterative algorithm will converge to the global minimum-error solution from any initial guess for the rotation. There is no need to sample the space of rotations in this case.

Also, instead of sampling the space of rotations in a systematic way using a finite group of rotations, one can generate points randomly distributed on the surface of the unit sphere in four-dimensional space. This provides a simpler means of generating initial guesses, although more initial guesses have to be tried than when a systematic procedure is used, since the space of rotations will not be sampled evenly.

The method as presented minimizes the sum of the squares of the weighted triple products $[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r]$. We assumed that the weighting factors vary slowly during the iterative process, so that we can use the current estimates of the baseline and rotation in computing the weighting factors. That is, when taking derivatives, the weighting factors are treated as constants. This is a good approximation when the parameters vary slowly, as they will when one is close to a minimum.

The method described above can be interpreted as a straightforward *weighted least-squares optimization*, which does not allow estimation of uncertainty in the parameters. One can also apply more sophisticated analyses to this problem, such as *best linear unbiased estimation*, which does not require any assumptions to be made about the distribution of the errors, only that their standard deviations be known. The standard deviations of the resultant parameters can then be used to evaluate their uncertainty, although no testing of confidence intervals is possible. Finally, one may apply *maximum likelihood estimation* of the orientation parameters,

where the observation errors are assumed to be distributed in a Gaussian fashion with known standard deviations. This allows one to derive confidence regions for the estimate orientation parameters, which can be treated as quantities that contain an error that is distributed in Gaussian fashion also.

12 Critical Surfaces

In certain rare cases, relative orientation cannot be accurately recovered, even when there are five or more pairs of rays. Normally, each error term varies linearly with distance in parameter space from the location of an extremum, and so the sum of squares of errors varies quadratically. There are situations, however, where the error terms do not vary linearly with distance, but quadratically or higher order, in certain special directions in parameter space. In this case, the sum of squares of errors does not vary quadratically with distance from the extremum, but as a function of the fourth or even higher power of this distance. This makes it very dificult to accurately locate the extremum. In this case, the total error is not significantly affected by a change in the rotation, as long as this change is accompanied by an appropriate corresponding change in the baseline. It turns out that this problem arises only when the observed scene points lie on certain surfaces called *Gefährliche Flächen* or *critical surfaces* [2, 18, 43, 56]. We show next that only points on certain hyperboloids of one sheet and their degenerate forms can lead to this kind of problem.

We could try to find a direction of movement in parameter space ($\delta\mathbf{b}$, $\delta\omega$) that leaves the total error unaffected (to second order) when given a particular surface. Instead, we will take the critical direction of motion in the parameter space as given, and try to find a surface for which the total error does not change (to second order).

Let $\mathbf{R}$ be a point on the surface, measured in the right camera coordinate system. Then

$$\beta\mathbf{r}_r = \mathbf{R} \quad \text{and} \quad \alpha\mathbf{r}_l' = \mathbf{b} + \mathbf{R} \tag{68}$$

for some positive α and β. In the absence of measurement errors,

$$[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] = \frac{1}{\alpha\beta}[\mathbf{b}\ (\mathbf{b} + \mathbf{R})\ \mathbf{R}] = 0 \tag{69}$$

We noted earlier that when we change the base line and the rotation slightly, the triple product $[\mathbf{b}\ \mathbf{r}_l'\mathbf{r}_r]$ becomes

$$[(\mathbf{b} + \delta\mathbf{b})\ (\mathbf{r}_l' + \delta\omega \times \mathbf{r}_l')\ \mathbf{r}_r] \tag{70}$$

or, if we ignore higher-order terms.

$$[\mathbf{b}\ \mathbf{r}_l'\ \mathbf{r}_r] + (\mathbf{r}_l' \times \mathbf{r}_r) \cdot \delta\mathbf{b} + (\mathbf{r}_r \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{r}_l') \tag{71}$$

The problem we are focusing on here arises when this error term is unchanged (to second order) for small movement in some direction in the parameter space. That is when

$$(\mathbf{r}_l' \times \mathbf{r}_r) \cdot \delta\mathbf{b} + (\mathbf{r}_r \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{r}_l') = 0 \tag{72}$$

for some $\delta\mathbf{b}$ and $\delta\omega$. Introducing the coordinates of the imaged points we obtain

$$\frac{1}{\alpha\beta}\{[(\mathbf{b} + \mathbf{R}) \times \mathbf{R}) \cdot \delta\mathbf{b} + (\mathbf{R} \times \mathbf{b}) \cdot (\delta\omega \times (\mathbf{b} + \mathbf{R})]\} = 0 \tag{73}$$

or

$$(\mathbf{R} \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{R}) + (\mathbf{R} \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{b}) + [\mathbf{b}\ \mathbf{R}\ \delta\mathbf{b}] = 0 \tag{74}$$

If we expand the first of the dot-products of the cross-products, we can write this equation in the form

$$(\mathbf{R} \cdot \mathbf{b})(\delta\omega \cdot \mathbf{R}) - (\mathbf{b} \cdot \delta\omega)(\mathbf{R} \cdot \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} = 0 \tag{75}$$

where

$$\mathbf{L} = \ell \times \mathbf{b}, \quad \text{while} \quad \ell = \mathbf{b} \times \delta\omega + \delta\mathbf{b} \tag{76}$$

The expression on the left-hand side contains a part that is quadratic in $\mathbf{R}$ and a part that is linear. The expression is clearly quadratic in X, Y, and Z, the components of the vector $\mathbf{R} = (X, Y, Z)^T$. Thus a surface leading to the kind of problem described above must be a quadric surface [26].

Note that there is not constant term in the equation of the surface, so $\mathbf{R} = \mathbf{0}$ satisfies equation (75). This means that the surface passes through the right projection center. It is easy to verify that $\mathbf{R} = -\mathbf{b}$ satisfies the equation also, which means that the surface passes through the left projection center as well. In fact, the whole baseline (and its extensions), $\mathbf{R} = k\mathbf{b}$, lies in the surface. This means that we must be dealing with a ruled quadric surface. It can consequently not be an ellipsoid or hyperboloid of two sheets, or one of their degenerate forms. The surface must be a hyperboloid of one sheet, or one of its degenerate forms. Additional information about the properties of these surfaces is

given in appendix B, while the degenerate forms are explored in appendix C (see also [33]).

It should be apparent that this kind of ambiguity is quite rare. This is nevertheless an issue of practical importance, since the accuracy of the solution is reduced if the points lie *near* some critical surface. A textbook case of this occurs in aerial photography of a roughly U-shaped valley taken along a flight line parallel to the axis of the valley from a height above the valley floor approximately equal to the width of the valley. In this case, the surface can be approximated by a portion of a circular cylinder with the baseline lying on the cylinder. This means that it is close to one of the degenerate forms of the hyperboloid of one sheet (see appendix C).

Note that hyperboloids of one sheet and their degenerate forms are exactly the surfaces that lead to ambiguity in the case of motion vision. The coordinate systems and symbols have been chosen here to make the correspondence between the two problems more apparent. The relationship between the two situations is nevertheless not quite as transparent as I had thought at first [21].

In the case of the ambiguity of the motion field, we are dealing with a two-way ambiguity arising from *infinitesimal* displacements in camera position and orientation. In the case of relative orientation, on the other hand, we are dealing with an elongated region in parameter space within which the error varies more slowly than quadratically, arising from images taken with cameras that have *finite* differences in position and orientation. Also note that the symbol $\delta\omega$ stands for a small change in a finite rotation here, while it refers to a difference in instantaneous rotational velocities in the motion vision case.

In practice, the relative orientation problem becomes ill conditioned near a solution that corresponds to ray intersections that lie close to a critical surface. In this case the surfaces of constant error in parameter space become very elongated and the location of the true minimum is not well defined. In addition, iterative algorithms based on local linearization tend to require many steps for convergence in this situation. It is important to point out that a given pair of bundles of corresponding rays may lead to poor behavior near one particular solution, yet be perfectly well behaved near other solutions. In general these sorts of problems are more likely to be found when the fields of view of one or both cameras are small. It is possible, however, to have ill-conditioned problems with wide fields of view. Conversely, a small field of view does not automatically lead to poor behavior.

Difficulties are also encountered when two local minima are near one another, since the surfaces of constant error in this case tend to be elongated along the direction in parameter space connecting the two minima and there is a saddle point somewhere between the two minima. At the saddle point the normal matrix is likely to be singular.

13 Conclusions

Methods for recovering the relative orientation of two cameras are of importance in both binocular stereo and motion vision. A new iterative method for finding the relative orientation has been described here. It can be used even when there is no initial guess available for the rotation or the baseline. The new method does not use Euler angles to represent the orientation and it does not require that the measured points be arranged in a particular pattern, as some previous methods do.

When there are many pairs of corresponding rays, the iterative method finds the global minimum from any starting point in parameter space. Local minima in the sum of squares of errors occur, however, when there are relatively few pairs of corresponding rays available. Methods for efficiently locating the global minimum in this case were discussed. When only five pairs of corresponding rays are given, several exact solutions of the coplanarity equations can be found. Typically only one of these is a positive solution, that is, one that yields positive distances to all the points in the scene. This allows one to pick the correct solution even when there is no initial guess available.

The solution cannot be determined with accuracy when the scene points lie on a critical surface.

Acknowledgments

The author thanks W. Eric L. Grimson and Rodney A. Brooks, who made helpful comments on a draft of this paper, as well as Michael Gennert who insisted that a least-squares procedure should be based on the error in determining the direction of the rays, not on the distance of their closest approach. I would also like to

thank Shahriar Negadahripour, who worked on the application of results developed earlier for critical surfaces in motion vision to the case when the step between successive images is relatively large. Harpreet Sawhney brought the recent paper by Faugeras and Maybank to my attention, while S. (Kicha) Ganapathy alerted me to the work of Netravali et al.

Finally, I am very grateful to the anonymous reviewers for pointing out several relevant references, including Pope [37], Hinsken [16, 17], and Longuett-Higgins [29], and for making significant contributions to the presentation. In particular, one of the reviewers drew my attention to the fact that one can solve the symmetric normal equations directly, without first eliminating the Lagrange multiplier. This not only simplifies the presentation, but leads to a (small) reduction in computational effort.

References

1. L.U. Bender, "Derivation of parallax equations," *Photogrammetric Engineering* 33 (10): 1175–1179, 1967.
2. A. Brandenberger, "Fehlertheorie der äusseren Orientierung von Steilaufnahmen," Ph.D. Thesis, Eidgenössische Technische Hochschule, Zürich, Switzerland, 1947.
3. P. Brou, "Using the Gaussian image to find the orientation of an object," *Intern. J. Robotics Res.* 3 (4): 89–125, 1983.
4. A.R. Bruss and B.K.P. Horn, "Passive navigation," *Comput. Vision, Graphics, Image Process.* 21 (1): 3–20, 1983.
5. O.D. Faugeras and S. Maybank, "Motion from point matches: Multiplicity of solutions," *Proc. IEEE Workshop Motion Vision*, March 20–22, 1989.
6. W. Förstner, "Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision," *Comput. Vision, Graphics, Image Process.* 40 (3): 273–310, 1987.
7. R.B. Forrest, "AP-C plotter orientation," *Photogrammetric Engineering* 32 (5): 1024–1027, 1966.
8. C.S. Fraser and D.C. Brown, "Industrial photogrammetry: New developments and recent applications," *Photogrammetric Record* 12 (68): 197–217, October 1986.
9. M.A. Gennert, "A computational framework for understanding problems in stereo vision." Ph.D. thesis, Department of Electrical Engineering and Computer Science, MIT, August 1987.
10. C.D. Ghilani, "Numerically assisted relative orientation of the Kern PG-2," *Photogramm. Engineer. Remote Sensing* 49 (10): 1457–1459, 1983.
11. S.K. Ghosh, "Relative orientation improvement," *Photogrammetric Engineering* 32 (3): 410–414, 1966.
12. S.K. Ghosh, *Theory of Stereophotogrammetry*. Ohio University Bookstores: Columbus, OH, 1972.
13. C. Gill, "Relative orientation of segmented, panoramic grid models on the AP-II," *Photogrammetric Engineering* 30: 957–962, 1964.
14. B. Hallert, *Photogrammetry*. McGraw-Hill: New York, 1960.
15. D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination*. Chelsea Publishing: New York, 1953, 1983.
16. L. Hinsken, "Algorithmen zur Beschaffung von Näherungswerten für die Orientierung von beliebig im Raum angeordneten Strahlenbündeln," Dissertation, Deutsche Geodätische Kommission, Reihe C, Heft Nr. 333, München, Federal Republic of Germany, 1987.
17. L. Hinsken, "A singularity-free algorithm for spatial orientation of bundles," *Intern. Arch. Photogramm. Remote Sensing* 27, B5, comm. v, pp. 262–272, 1988.
18. W. Hofmann, "Das Problem der 'Gefährlichen Flächen' in Theorie und Praxis," Ph.D. Thesis, Technische Hochschule München. Published in 1953 by Deutsche Geodätische Kommission, München, Federal Republic of Germany, 1949.
19. B.K.P. Horn, *Robot Vision*. MIT Press: Cambridge, MA, and McGraw-Hill, New York, 1986.
20. B.K.P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am.* A 4 (4): 629–642, 1987.
21. B.K.P. Horn, "Motion fields are hardly ever ambiguous," *Intern. J. Comput. Vision* 1 (3): 263–278, 1987.
22. B.K.P. Horn, "Relative orientation," Memo 994, Artificial Intelligence Laboratory, MIT, Cambridge, MA, 1987. Also, *Proc. Image Understanding Workshop*, 6–8 April, Morgan Kaufman Publishers: San Mateo, CA, pp. 826–837, 1988.
23. B.K.P. Horn, H.M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Am.* A 5 (7): 1127–1135, July 1988.
24. B.K.P. Horn and E.J. Weldon Jr., "Direct methods for recovering motion," *Intern. J. Comput. Vision* 2 (1): 51–76, June 1988.
25. H. Jochmann, "Number of orientation points," *Photogrammetric Engineering* 31 (4): 670–679, 1965.
26. G.A. Korn and T.M. Korn, *Mathematical Handbook for Scientists and Engineers*, 2nd ed. McGraw-Hill: New York, 1968.
27. H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature* 293: 133–135, September 1981.
28. H.C. Longuet-Higgins, "The reconstruction of a scene from two projections—Configurations that defeat the eight-point algorithm," *IEEE, Proc. 1st Conf. Artif. Intell. Applications*, Denver, CO, 1984.
29. H.C. Longuet-Higgins, "Multiple interpretations of a pair of images of a surface," *Proc. Roy. Soc. London* 418: 1–15, 1988.
30. H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Roy. Soc. London* B 208: 385–397, 1980.
31. E.M. Mikhail and F. Ackerman, *Observations and Least Squares*. Harper & Row: New York, 1976.
32. F. Moffit and E.M. Mikhail, *Photogrammetry*, 3rd ed. Harper & Row: New York, 1980.
33. S. Negahdaripour, "Multiple interpretation of the shape and motion of objects from two perspective images." Unpublished manuscript of the Waikiki Surfing and Computer Vision Society of the Department of Electrical Engineering at the University of Hawaii at Manoa, Honolulu, HI, 1989.
34. A.N. Netravali, T.S. Huang, A.S. Krisnakumar, and R.J. Holt, "Algebraic methods in 3-D motion estimation from two-view point correspondences." Unpublished internal report, A.T.&T. Bell Laboratories: Murray Hill, NJ, 1989.

35. A. Okamoto, "Orientation and construction of models—Part I: The orientation problem in close-range photogrammetry," *Photogramm. Engineer. Remote Sensing* 47 (10): 1437–1454, 1981.
36. H.L. Oswal, "Comparison of elements of relative orientation," *Photogrammetric Engineering* 33 (3): 335–339, 1967.
37. A. Pope, "An advantageous, alternative parameterization of rotations for analytical photogrammetry," ESSA Technical Report CaGS-39, Coast and Geodetic Survey, U.S. Department of Commerce, Rockville, MA. Also *Symposium on Computational Photogrammetry.* American Society of Photogrammetry: Alexandria, Virginia, January 7–9, 1970.
38. K. Rinner, "Studien über eine allgemeine, vorraussetzungslose Lösung des Folgebildanschlusses," *Österreichische Zeitschrift für Vermessung*, Sonderheft 23, 1963.
39. S. Sailor, "Demonstration board for stereoscopic plotter orientation," *Photogrammetric Engineering* 31 (1): 176–179, January, 1965.
40. E. Salamin, "Application of quaternions to computation with rotations." Unpulbished Internal Report, Stanford University, Stanford, CA, 1979.
41. G.H. Schut, "An analysis of methods and results in analytical aerial triangulation," *Photogrammetria* 14: 16–32, 1957–1958.
42. G.H. Schut, "Construction of orthogonal matrices and their application in analytical photogrammetry," *Photogrammetria* 15 (4): 149–162, 1958–1959.
43. K. Schwidefsky, *An Outline of Photogrammetry.* Translated by John Fosberry, 2nd ed. Pitman & Sons: London, 1973.
44. K. Schwidefsky, and F. Ackermann, *Photogrammetrie.* Teubner: Stuttgart, Federal Republic of Germany, 1976.
45. C.C. Slama, C. Theurer, and S.W. Hendrikson (eds.), *Manual of Photogrammetry.* American Society of Photogrammetry: Falls Church, VA, 1980.
46. J.H. Stuelpnagle, "On the parameterization of the three-dimensional rotation group," *SIAM Review* 6 (4): 422–430, October 1964.
47. R.H. Taylor, "Planning and execution of straight line manipulator trajectories." In *Robot Motion: Planning and Control*, M.J. Brady, J.M. Hollerbach, T.L. Johnson, R. Lozano-Pérez, and M.T. Mason (eds.), MIT Press: Cambridge, MA, 1982.
48. E.H. Thompson, "A Method for the construction of orthonormal matrices," *Photogrammetric Record* 3 (13): 55–59, April 1959.
49. E.H. Thompson, "A rational algebraic formulation of the problem of relative orientation," *Photogrammetric Record* 3 (14): 152–159, 1959.
50. E.H. Thompson, "A note on relative orientation," *Photogrammetric Record* 4 (24): 483–488, 1964.
51. E.H. Thompson, "The projective theory of relative orientation," *Photogrammetria* 23: 67–75, 1968.
52. R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. PAMI* 6 (1): 13–27, 1984.
53. S. Ullman, *The Interpretation of Visual Motion.* MIT Press: Cambridge, MA, 1979.
54. A.J. Van Der Weele, "The relative orientation of photographs of mountainous terrain," *Photogrammetria* 16 (2): 161–169, 1959–1960.
55. P.R. Wolf, *Elements of Photogrammetry*, 2nd ed. McGraw-Hill: New York, 1983.
56. M. Zeller, *Textbook of Photogrammetry.* H.K. Lewis & Company: London, 1952.

Appendix A: Rotation Groups of Regular Polyhedra

Each of the rotation groups of the regular polyhedra can be generated from two judiciously chosen elements. For convenience, however, an explicit representation of all of the elements of each of the groups is given here. The number of different component values occurring in the unit quaternions representing the rotations can be kept low by careful choice of the alignment of the polyhedron with the coordinate axes. The attitudes of the polyhedra here were selected to minimize the number of different numerical values that occur in the components of the unit quaternions. A different representation of the group is obtained if the vector parts of each of the unit quaternions is rotated in the same way. This just corresponds to the rotation group of the polyhedron in a different attitude with respect to the underlying coordinate system. This observation leads to a convenient way of generating finer systematic sampling patterns of the space of rotations than the ones provided directly by the rotation group of a regular polyhedron in a particular alignment with the coordinate axes (see also [3]).

The components of the unit quaternions here may take on the values 0 and 1, as well as the following:

$$a = \frac{\sqrt{5}-1}{4}, \quad b = \frac{1}{2}, \quad c = \frac{1}{\sqrt{2}}, \quad \text{and} \quad d = \frac{\sqrt{5}+1}{4} \tag{77}$$

Here are the unit quaternions for the twelve elements of the rotation group of the tetrahedron:

$$\begin{array}{llll} (1,0,0,0) & (0,1,0,0) & (0,0,1,0) & (0,0,0,1) \\ (b,b,b,b) & (b,b,b,-b) & (b,b,-b,b) & (b,b,-b,-b) \\ (b,-b,b,b) & (b,-b,b,-b) & (b,-b,-b,b) & (b,-b,-b,-b) \end{array} \tag{78}$$

Here are the unit quaternions for the twenty-four elements of the rotation group of the octahedron and the hexahedron (cube):

$$\begin{array}{llll} (1,0,0,0) & (0,1,0,0) & (0,0,1,0) & (0,0,0,1) \\ (0,0,c,c) & (0,0,c,-c) & (0,c,0,c) & (0,c,0,-c) \\ (0,c,c,0) & (0,c,-c,0) & (c,0,0,c) & (c,0,0,-c) \\ (c,0,c,0) & (c,0,-c,0) & (c,c,0,0) & (c,-c,0,0) \\ (b,b,b,b) & (b,b,b,-b) & (b,b,-b,b) & (b,b,-b,-b) \\ (b,-b,b,b) & (b,-b,b,-b) & (b,-b,-b,b) & (b,-b,-b,-b) \end{array} \tag{79}$$

Here are the unit quaternions for the sixty elements of the rotation group of the icosahedron and the dodecahedron:

(1, 0, 0, 0)	(0, 1, 0, 0)	(0, 0, 1, 0)	(0, 0, 0, 1)
(0, *a*, *b*, *d*)	(0, *a*, *b*, −*d*)	(0, *a*, −*b*, *d*)	0, *a*, −*b*, −*d*)
(0, *b*, *d*, *a*)	(0, *b*, *d*, −*a*)	(0, *b*, −*d*, *a*)	(0, *b*, −*d*, −*a*)
(0, *d*, *a*, *b*)	(0, *d*, *a*, −*b*)	(0, *d*, −*a*, *b*)	(0, *d*, −*a*, −*b*)
(*a*, 0, *d*, *b*)	(*a*, 0, *d*, −*b*)	(*a*, 0, −*d*, *b*)	(*a*, 0, −*d*, −*b*)
(*b*, 0, *a*, *d*)	(*b*, 0, *a*, −*d*)	(*b*, 0, −*a*, *d*)	(*b*, 0, −*a*, −*d*)
(*d*, 0, *b*, *a*)	(*d*, 0, *b*, −*a*)	(*d*, 0, −*b*, *a*)	(*d*, 0, −*b*, −*a*)
(*a*, *b*, 0, *d*)	(*a*, *b*, 0, −*d*)	(*a*, −*b*, 0, *d*)	(*a*, −*b*, 0, −*d*)
(*b*, *d*, 0, *a*)	(*b*, *d*, 0, −*a*)	(*b*, −*d*, 0, *a*)	(*b*, −*d*, 0, −*a*)
(*d*, *a*, 0, *b*)	(*d*, *a*, 0, −*b*)	(*d*, −*a*, 0, *b*)	(*d*, −*a*, 0, −*b*)
(*a*, *d*, *b*, 0)	(*a*, *d*, −*b*, 0)	(*a*, −*d*, *b*, 0)	(*a*, −*d*, −*b*, 0)
(*b*, *a*, *d*, 0)	(*b*, *a*, −*d*, 0)	(*b*, −*a*, *d*, 0)	(*b*, −*a*, −*d*, 0)
(*d*, *b*, *a*, 0)	(*d*, *b*, −*a*, 0)	(*d*, −*b*, *a*, 0)	(*d*, −*b*, −*a*, 0)
(*b*, *b*, *b*, *b*,)	(*b*, *b*, *b*, −*b*)	(*b*, *b*, −*b*, *b*)	(*b*, *b*, −*b*, −*b*)
(*b*, −*b*, *b*, *b*)	(*b*, −*b*, *b*, −*b*)	(*b*, −*b*, −*b*, *b*)	(*b*, −*b*, −*b*, −*b*)

(80)

Remember that changing the signs of all the components of a unit quaternion does not change the rotation that it represents.

Appendix B: Some Properties of Critical Surfaces

In this appendix we develop some more of the properties of the critical surfaces. The equation of a critical surface can be written in the form

$$(\mathbf{R} \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} = 0 \quad (81)$$

or

$$(\mathbf{R} \cdot \mathbf{b})(\delta\omega \cdot \mathbf{R}) - (\mathbf{b} \cdot \delta\omega)(\mathbf{R} \cdot \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} = 0 \quad (82)$$

where

$$\mathbf{L} = \ell \times \mathbf{b}, \quad \text{while} \quad \ell = \mathbf{b} \times \delta\omega + \delta\mathbf{b} \quad (83)$$

It is helpful to first establish some simple relationships between the quantities appearing in formula (83). We start with the observations that $\ell \cdot \mathbf{b} = 0$, that $\ell \cdot \delta\omega = \delta\mathbf{b} \cdot \delta\omega$, and $\ell \times \delta\mathbf{b} = -(\delta\mathbf{b} \cdot \delta\omega)\mathbf{b}$.

We can also expand $\mathbf{L}$ to yield,

$$\mathbf{L} = \delta\omega - (\mathbf{b} \cdot \delta\omega)\mathbf{b} + \delta\mathbf{b} \times \mathbf{b} \quad (84)$$

It follows that $\mathbf{L} \cdot \mathbf{b} = 0$; that $\mathbf{L} \cdot \delta\mathbf{b} = \delta\omega \cdot \delta\mathbf{b}$, and

$$\begin{aligned} \mathbf{L} \times \mathbf{b} &= -(\mathbf{b} \times \delta\omega + \delta\mathbf{b}) = -\ell\,, \\ \mathbf{L} \times \delta\omega &= (\delta\mathbf{b} \cdot \delta\omega)\mathbf{b} - (\mathbf{b} \cdot \delta\omega)\ell \end{aligned} \quad (85)$$

We have already established that $\mathbf{R} = k\mathbf{b}$ is an equation for one of the rulings passing through the origin. A hyperboloid of one sheet has two intersecting families of rulings, so there should be a second ruling passing through the origin. Consider the vector $\mathbf{S}$ defined by

$$\mathbf{S} = (\mathbf{L} \times \delta\omega) \times \mathbf{L} \quad (86)$$

which can be written in the form

$$\mathbf{S} = (\mathbf{L} \cdot \mathbf{L})\delta\omega - (\mathbf{L} \cdot \delta\omega)\mathbf{L} \quad (87)$$

or

$$\mathbf{S} = (\delta\omega \cdot \delta\mathbf{b})\ell + (\mathbf{b} \cdot \delta\omega)(\ell \cdot \ell)\mathbf{b} \quad (88)$$

so that $\mathbf{S} \cdot \mathbf{b} = (\mathbf{b} \cdot \delta\omega)(\ell \cdot \ell)$ and $\mathbf{S} \cdot \delta\omega = (\delta\omega \cdot \delta\mathbf{b})^2 + (\mathbf{b} \cdot \delta\omega)^2(\ell \cdot \ell)$.

If we substitute $\mathbf{R} = k\mathbf{S}$ into the formula

$$(\mathbf{R} \times \mathbf{b}) \cdot (\delta\omega \times \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} \quad (89)$$

we obtain zero, since $\mathbf{L} \cdot \mathbf{S} = 0$ and

$$\mathbf{S} \times \mathbf{b} = (\delta\omega \cdot \delta\mathbf{b})\mathbf{L} \quad (90)$$

is orthogonal to

$$\mathbf{S} \times \delta\omega = -(\mathbf{L} \cdot \delta\omega)\mathbf{L} \times \delta\omega \quad (91)$$

We conclude that $\mathbf{R} = k\mathbf{S}$ is an equation for the other ruling that passes through the right projection center.

There are two families of parallel planes that cut an ellipsoid in circular cross-sections [15]. Similarly, there are two families of parallel planes that cut a hyperboloid of one sheet in circular cross-sections. One of these families consists of planes perpendicular to the baseline, that is, with common normal $\mathbf{b}$. We can see this by substituting $\mathbf{R} \cdot \mathbf{b} = k$ in the equation of the critical surface. We obtain

$$k(\delta\omega \cdot \mathbf{R}) - (\mathbf{b} \cdot \delta\omega)(\mathbf{R} \cdot \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} = 0 \quad (92)$$

or

$$(\mathbf{b} \cdot \delta\omega)(\mathbf{R} \cdot \mathbf{R}) - (k\delta\omega + \mathbf{L}) \cdot \mathbf{R} = 0 \quad (93)$$

This is the equation of a sphere, since the only second-order term in $\mathbf{R}$ is a multiple of

$$\mathbf{R} \cdot \mathbf{R} = X^2 + Y^2 + Z^2 \quad (94)$$

We can conclude that the intersection of the critical surface and the plane is also the intersection of this sphere and the plane, and so must be a circle. The same applies to the intersection of the critical surface and the family of planes with common normal $\delta\omega$, since we get

$$(\mathbf{b} \cdot \delta\omega)(\mathbf{R} \cdot \mathbf{R}) - (k\mathbf{b} + \mathbf{L}) \cdot \mathbf{R} = 0 \quad (95)$$

when we substitute $\mathbf{R} \cdot \delta\omega = k$ into the equation of the critical surface.

The equation of the critical surface is given in the implicit form $f(\mathbf{R}) = 0$. The equation of a tangent plane to such a surface can be obtained by differentiating with respect to $\mathbf{R}$:

$$\mathbf{N} = (\mathbf{R} \times \delta\omega) \times \mathbf{b} + (\mathbf{R} \times \mathbf{b}) \times \delta\omega + \mathbf{L} \quad (96)$$

or

$$\mathbf{N} = (\mathbf{R} \cdot \mathbf{b})\delta\omega + (\mathbf{R} \cdot \delta\omega)\mathbf{b} - 2(\mathbf{b} \cdot \delta\omega)\mathbf{R} + \mathbf{L} \quad (97)$$

The tangent plane at the origin has normal $\mathbf{L}$. This tangent plane contains the baseline (since $\mathbf{L} \cdot \mathbf{b} = 0$), as well as the other ruling passing through the origin (since $\mathbf{L} \cdot \mathbf{S} = 0$). Note that the normal to the tangent plane is not constant along either of these rulings, as they would be if we were dealing with a developable surface.

In the above we have not considered a large number of degenerate situations that can occur. The reader is referred to appendix C for a detailed analysis of these.

Appendix C: Degenerate Critical Surfaces

There are a number of special alignments of the infinitesimal change in the rotation, $\delta\omega$, with the baseline, $\mathbf{b}$, and the infinitesimal change in the baseline $\delta\mathbf{b}$ that lead to degenerate forms of the hyperboloid of one sheet.

One of the rulings passing through the origin is given by $\mathbf{R} = k\mathbf{b}$, while the other is given by $\mathbf{R} = k\mathbf{S}$. If these two rulings become parallel, we are dealing with a degenerate form that has only one set of rulings, that is a conical surface. Now

$$\mathbf{S} = (\delta\omega \cdot \delta\mathbf{b})\ell + (\mathbf{b} \cdot \delta\omega)(\ell \cdot \ell)\mathbf{b} \quad (98)$$

is parallel to $\mathbf{b}$ only when $(\delta\omega \cdot \delta\mathbf{b}) = 0$, since ℓ is perpendicular to $\mathbf{b}$. In this case

$$\delta\mathbf{b} \cdot \delta\omega = 0 \quad \text{and} \quad \delta\mathbf{b} \cdot \mathbf{b} = 0 \quad (99)$$

so $\delta\mathbf{b} = k(\mathbf{b} \times \delta\omega)$ for some constant k. Consequently $\ell = (k + 1)(\mathbf{b} \times \delta\omega)$. The vertex of the conical surface must lie on the baseline since the baseline is a ruling, and every ruling passes through the vertex. It can be shown that the vertex actually lies at $\mathbf{R} = -(k + 1)\mathbf{b}$.

We also know that cross-sections in planes perpendicular to the baseline are circles. This tells us that we are dealing with elliptical cones. Right circular cones cannot be critical surfaces. It can be shown that the main axis of the elliptical cone lies in the direction $\mathbf{b} + \delta\omega$.

A special case of the special case above occurs when

$$\|\mathbf{b} \times \delta\omega\| = 0 \quad (100)$$

that is $\delta\omega \| \mathbf{b}$. Here $\delta\omega = k\mathbf{b}$ for some constant k and so $\ell = \delta\mathbf{b}$ and $\mathbf{L} = \delta\mathbf{b} \times \mathbf{b}$. The equation of the surface becomes

$$k(\mathbf{R} \cdot \mathbf{b})^2 - k(\mathbf{b} \cdot \mathbf{b})(\mathbf{R} \cdot \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} = 0 \quad (101)$$

or

$$k\|\mathbf{R} \times \mathbf{b}\|^2 + (\delta\mathbf{b} \times \mathbf{b}) \cdot \mathbf{R} = 0 \quad (102)$$

This is the equation of a circular cylinder with axis parallel to the baseline. In essence, the vertex of the cone has receded to infinity along the baseline.

Another special case arises when the radius of the circular cross-sections with planes perpendicular to the baseline becomes infinite. In this case we obtain straight lines, and hence rulings, in these planes. The hyperbolic paraboloid is the degenerate form that has the property that each of its two sets of rulings can be obtained by cutting the surface with a set of parallel planes [15]. This happens when $\delta\omega$ is perpendicular to $\mathbf{b}$, that is, $\mathbf{b} \cdot \delta\omega = 0$. The equation of the surface in this case simplifies to

$$(\mathbf{R} \cdot \mathbf{b})(\delta\omega \cdot \mathbf{R}) + \mathbf{L} \cdot \mathbf{R} = 0. \quad (103)$$

The intersection of this surface with any plane perpendicular to the baseline is a straight line. We can show this by substituting

$$\mathbf{R} \cdot \mathbf{b} = k \quad (104)$$

into the equation of the surface. We obtain

$$(k\,\delta\omega + \mathbf{L}) \cdot \mathbf{R} = 0 \quad (105)$$

that is, the equation of another plane. Now the intersection of two planes is a straight line. So we may conclude that the intersection of the surface and the original plane is a straight line. We can show in the same way that the intersection of the surface with any plane perpendicular to $\delta\omega$ is a straight line by substituting

$$\mathbf{R} \cdot \delta\omega = k \quad (106)$$

into the equation of the surface. It can be shown that the saddle point of the hyperbolic paraboloid surface lies on the baseline.

A special case of particular interest arises when $\delta\omega$ is perpendicular to both $\mathbf{b}$ and $\delta\mathbf{b}$, that is,

$$\mathbf{b} \cdot \delta\omega = 0 \quad \text{and} \quad \delta\mathbf{b} \cdot \delta\omega = 0 \quad (107)$$

and so $\delta\omega = k(\delta\mathbf{b} \times \mathbf{b})$, for some constant k. Then $\ell = (k + 1)\,\delta\mathbf{b}$ and $\mathbf{L} = (k + 1)(\delta\mathbf{b} \times \mathbf{b})$. The equation of the surface becomes

$$k(\mathbf{R} \cdot \mathbf{b})[(\delta\mathbf{b} \times \mathbf{b}) \cdot \mathbf{R}] + (k + 1)[(\delta\mathbf{b} \times \mathbf{b}) \cdot \mathbf{R}] = 0 \quad (108)$$

or just

$$[(k(\mathbf{R} \cdot \mathbf{b}) + (k + 1)][(\delta\mathbf{b} \times \mathbf{b}) \cdot \mathbf{R}] = 0 \quad (109)$$

so either

$$(\delta\mathbf{b} \times \mathbf{b}) \cdot \mathbf{R} = 0 \quad \text{or} \quad k(\mathbf{R} \cdot \mathbf{b}) + (k + 1) = 0 \quad (110)$$

The first of these is the equation of a plane containing the baseline $\mathbf{b}$ and the vector $\delta\mathbf{b}$. The second is the equation of a plane perpendicular to the baseline. So the solution degenerates in this case into a surface consisting of two intersecting planes. One of these planes appears only as a line in each of the two images, since it passes through both projection centers, and so does not really contribute to the image. It is fortunate that planes can only be degenerate surfaces if they are perpendicular to the baseline, since surfaces that are almost planar occur frequently in aerial photography.[22]

To summarize then, we have the following degenerate cases:

- elliptical cones when $\delta\omega \perp \delta\mathbf{b}$,
- circular cylinders when $\delta\omega \parallel \mathbf{b}$
- hyperbolic paraboloids when $\delta\omega \perp \mathbf{b}$, and
- intersecting planes when $\delta\omega \perp \delta\mathbf{b}$ and $\delta\omega \perp \mathbf{b}$.

For further details, and a proof that not all hyperboloids of one sheet passing through the origin lead to critical surfaces, see [21].

[22]The baseline was nearly perpendicular to the surface in the sequence of photographs obtained by the Ranger spacecraft as it crashed into the lunar surface. This made photogrammetric analysis difficult.