Graphics and Image Processing

J. D. Foley Editor

Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography

Martin A. Fischler and Robert C. Bolles
SRI International

A new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data is introduced. RANSAC is capable of interpreting/smoothing data containing a significant percentage of gross errors, and is thus ideally suited for applications in automated image analysis where interpretation is based on the data provided by error-prone feature detectors. A major portion of this paper describes the application of RANSAC to the Location Determination Problem (LDP): Given an image depicting a set of landmarks with known locations, determine that point in space from which the image was obtained. In response to a RANSAC requirement, new results are derived on the minimum number of landmarks needed to obtain a solution, and algorithms are presented for computing these minimum-landmark solutions in closed form. These results provide the basis for an automatic system that can solve the LDP under difficult viewing and analysis conditions. Implementation details and computational examples are also presented.

Key Words and Phrases: model fitting, scene analysis, camera calibration, image matching, location determination, automated cartography.

CR Categories: 3.60, 3.61, 3.71, 5.0, 8.1, 8.2

I. Introduction

We introduce a new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data; and illustrate its use in scene analysis and automated cartography. The application discussed, the location determination problem (LDP), is treated at a level beyond that of a mere example of the use of the RANSAC paradigm; new basic findings concerning the conditions under which the LDP can be solved are presented and a comprehensive approach to the solution of this problem that we anticipate will have near-term practical applications is described.

To a large extent, scene analysis (and, in fact, science in general) is concerned with the interpretation of sensed data in terms of a set of predefined models. Conceptually, interpretation involves two distinct activities: First, there is the problem of finding the best match between the data and one of the available models (the classification problem); Second, there is the problem of computing the best values for the free parameters of the selected model (the parameter estimation problem). In practice, these two problems are not independent—a solution to the parameter estimation problem is often required to solve the classification problem.

Classical techniques for parameter estimation, such as least squares, optimize (according to a specified objective function) the fit of a functional description (model) to *all* of the presented data. These techniques have no internal mechanisms for detecting and rejecting gross errors. They are averaging techniques that rely on the assumption (the smoothing assumption) that the maximum expected deviation of any datum from the assumed model is a direct function of the size of the data set, and thus regardless of the size of the data set, there will always be enough good values to smooth out any gross deviations.

In many practical parameter estimation problems the smoothing assumption does not hold; i.e., the data contain uncompensated gross errors. To deal with this situation, several heuristics have been proposed. The technique usually employed is some variation of first using all the data to derive the model parameters, then locating the datum that is farthest from agreement with the instantiated model, assuming that it is a gross error, deleting it, and iterating this process until either the maximum deviation is less then some preset threshold or until there is no longer sufficient data to proceed.

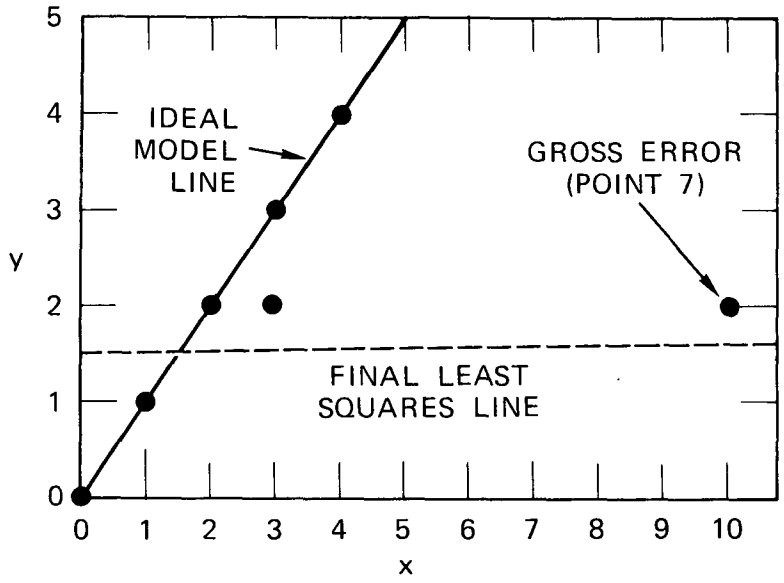
It can easily be shown that a single gross error ("poisoned point"), mixed in with a set of good data, can

The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract Nos. DAAG29-76-C-0057 and MDA903-79-C-0588.

Authors' Present Address: Martin A. Fischler and Robert C. Bolles, Artificial Intelligence Center, SRI International, Menlo Park CA 94025.

Fig. 1. Failure of Least Squares (and the "Throwing Out The Worst Residual" Heuristic), to Deal with an Erroneous Data Point.

PROBLEM: Given the set of seven (x,y) pairs shown in the plot, find a best fit line, assuming that no valid datum deviates from this line by more than 0.8 units.



POINT	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

COMMENT: Six of the seven points are valid data and can be fit by the solid line. Using Least Squares (and the "throwing out the worst residual" heuristic), we terminate after four iterations with four remaining points, including the gross error at (10,2) fit by the dashed line.

SUCCESSIVE LEAST SQUARES APPROXIMATIONS		
ITERATION	DATA SET	FITTING LINE
1	1, 2, 3, 4, 5, 6, 7	1.48 + .16x
2	1, 2, 3, 4, 5, 7	1.25 + .13x
3	1, 2, 3, 4, 7	0.96 + .14x
4	2, 3, 4, 7	1.51 + .06x

COMPUTATION OF RESIDUALS				
POINT	ITERATION 1 RESIDUALS	ITERATION 2 RESIDUALS	ITERATION 3 RESIDUALS	ITERATION 4 RESIDUALS
1	-1.48	-1.25	-.96*	—
2	-0.64	-0.38	-.10	-.57
3	-0.20	0.49	.76	.37
4	0.05	0.36	.63	.31
5	1.05	1.36*	—	—
6	1.89*	—	—	—
7	-1.06	-0.57	-.33	-.11

cause the above heuristic to fail (for example, see Figure 1). It is our contention that averaging is not an appropriate technique to apply to an unverified data set.

In the following section we introduce the RANSAC paradigm, which is capable of smoothing data that contain a significant percentage of gross errors. This paradigm is particularly applicable to scene analysis because local feature detectors, which often make mistakes, are the source of the data provided to the interpretation algorithms. Local feature detectors make two types of errors—classification errors and measurement errors. Classification errors occur when a feature detector incorrectly identifies a portion of an image as an occurrence of a feature. Measurement errors occur when the feature detector correctly identifies the feature, but slightly miscalculates one of its parameters (e.g., its image location). Measurement errors generally follow a normal distribution, and therefore the smoothing assumption is applicable to them. Classification errors, however, are gross errors, having a significantly larger effect than measurement errors, and do not average out.

In the final sections of this paper the application of RANSAC to the location determination problem is discussed:

Given a set of "landmarks" ("control points"), whose locations are known in some coordinate frame, determine the location (relative to the coordinate frame of the landmarks) of that point in space from which an image of the landmarks was obtained.

In response to a RANSAC requirement, some new results are derived on the minimum number of landmarks needed to obtain a solution, and then algorithms are presented for computing these minimum-landmark solutions in closed form. (Conventional techniques are iterative and require a good initial guess to assure convergence.) These results form the basis for an automatic system that can solve the LDP under severe viewing and analysis conditions. In particular, the system performs properly even if a significant number of landmarks are incorrectly located due to low visibility, terrain changes, or image analysis errors. Implementation details and experimental results are presented to complete our description of the LDP application.

II. Random Sample Consensus

The RANSAC procedure is opposite to that of conventional smoothing techniques: Rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the invalid data points, RANSAC uses as small an initial data set as feasible and enlarges this set with consistent data when possible. For example, given the task of fitting an arc of a circle to a set of two-dimensional points, the RANSAC approach would be to select a set of three points (since three points are required to determine a circle), compute the center and radius of the implied circle, and count the number of points that are close enough to that circle to suggest their compatibility with it (i.e., their deviations are small enough to be measurement errors). If there are enough compatible points, RANSAC would employ a smoothing technique such as least squares, to compute an improved estimate for the parameters of the circle now that a set of mutually consistent points has been identified.

The RANSAC paradigm is more formally stated as follows:

Given a model that requires a minimum of n data points to instantiate its free parameters, and a set of data points P such that the number of points in P is greater than n [$\#(P) \geq n$], randomly select a subset $S1$ of n data points from P and instantiate the model. Use the instantiated model $M1$ to determine the subset $S1^*$ of points in P that are within some error tolerance of $M1$. The set $S1^*$ is called the consensus set of $S1$.

If $\#(S1^*)$ is greater than some threshold t, which is a function of the estimate of the number of gross errors in P, use $S1^*$ to compute (possibly using least squares) a new model $M1^*$.

If $\#(S1^*)$ is less than t, randomly select a new subset $S2$ and repeat the above process. If, after some predetermined number of trials, no consensus set with t or more members has been found, either solve the model with the largest consensus set found, or terminate in failure.

There are two obvious improvements to the above algorithm: First, if there is a problem related rationale for selecting points to form the S's, use a deterministic selection process instead of a random one; second, once a suitable consensus set S^* has been found and a model M^* instantiated, add any new points from P that are consistent with M^* to S^* and compute a new model on the basis of this larger set.

The RANSAC paradigm contains three unspecified parameters: (1) the error tolerance used to determine whether or not a point is compatible with a model, (2) the number of subsets to try, and (3) the threshold t, which is the number of compatible points used to imply that the correct model has been found. Methods are discussed for computing reasonable values for these parameters in the following subsections.

A. Error Tolerance For Establishing Datum/Model Compatibility

The deviation of a datum from a model is a function of the error associated with the datum and the error associated with the model (which, in part, is a function of the errors associated with the data used to instantiate the model). If the model is a simple function of the data points, it may be practical to establish reasonable bounds on error tolerance analytically. However, this straightforward approach is often unworkable; for such cases it is generally possible to estimate bounds on error tolerance experimentally. Sample deviations can be produced by perturbing the data, computing the model, and measuring the implied errors. The error tolerance could then be set at one or two standard deviations beyond the measured average error.

The expected deviation of a datum from an assumed model is generally a function of the datum, and therefore the error tolerance should be different for each datum. However, the variation in error tolerances is usually

relatively small compared to the size of a gross error. Thus, a single error tolerance for all data is often sufficient.

B. The Maximum Number of Attempts to Find a Consensus Set

The decision to stop selecting new subsets of P can be based upon the expected number of trials k required to select a subset of n good data points. Let w be the probability that any selected data point is within the error tolerance of the model. Then we have:

$$E(k) = b + 2*(1 - b)*b + 3*(1 - b)^2*b \cdots + i*(1 - b)^{i-1}*b + \cdots,$$

$$E(k) = b*[1 + 2*a + 3*a^2 \cdots + i*a^{i-1} + \cdots],$$

where $E(k)$ is the expected value of k, $b = w^n$, and $a = (1 - b)$.
An identity for the sum of a geometric series is

$$a/(1 - a) = a + a^2 + a^3 \cdots + a^i + \cdots.$$

Differentiating the above identity with respect to a, we have:

$$1/(1 - a)^2 = 1 + 2*a + 3*a^2 \cdots + i*a^{i-1} + \cdots.$$

Thus,

$$E(k) = 1/b = w^{-n}$$

The following is a tabulation of some values of $E(k)$ for corresponding values of n and w:

w	$n = 1$	2	3	4	5	6
0.9	1.1	1.2	1.4	1.5	1.7	1.9
0.8	1.3	1.6	2.0	2.4	3.0	3.8
0.7	1.4	2.0	2.9	4.2	5.9	8.5
0.6	1.7	2.8	4.6	7.7	13	21
0.5	2.0	4.0	8.0	16	32	64
0.4	2.5	6.3	16	39	98	244
0.3	3.3	11	37	123	412	—
0.2	5.0	25	125	625	—	—

In general, we would probably want to exceed $E(k)$ trials by one or two standard deviations before we give up. Note that the standard deviation of k, $SD(k)$, is given by:

$$SD(k) = sqrt\,[E(k^2) - E(k)^2].$$

Then

$$E(k^2) = \sum_{i=0}^{\infty} (b*i^2*a^{i-1}),$$

$$= \sum_{i=0}^{\infty} [b*i*(i - 1)*a^{i-1}] + \sum_{i=0}^{\infty} (b*i*a^{i-1}),$$

but (using the geometric series identity and two differentiations):

$$2a/(1 - a)^3 = \sum_{i=0}^{\infty} (i*(i - 1)*a^{i-1}).$$

Thus,

$$E(k^2) = (2 - b)/(b^2),$$

and

$$SD(k) = [sqrt\,(1 - w^n)]*(1/w^n).$$

Note that generally $SD(k)$ will be approximately equal to $E(k)$; thus, for example, if $(w = 0.5)$ and $(n = 4)$, then $E(k) = 16$ and $SD(k) = 15.5$. This means that one might want to try two or three times the expected number of random selections implied by k (as tabulated above) to obtain a consensus set of more than t members.

From a slightly different point of view, if we want to ensure with probability z that at least one of our random selections is an error-free set of n data points, then we must expect to make at least k selections (n data points per selection), where

$$(1 - b)^k = (1 - z),$$

$$k = [\log(1 - z)]/[\log(1 - b)].$$

For example, if $(w = 0.5)$ and $(n = 4)$, then $(b = 1/16)$. To obtain a 90 percent assurance of making at least one error-free selection,

$$k = \log(0.1)/\log(15/16) = 35.7.$$

Note that if $w^n \ll 1$, then $k \approx \log(1 - z)E(k)$. Thus if $z = 0.90$ and $w^n \ll 1$, then $k \approx 2.3E(k)$; if $z = 0.95$ and $w^n \ll 1$, then $k \approx 3.0E(k)$.

C. A Lower Bound On the Size of an Acceptable Consensus Set

The threshold t, an unspecified parameter in the formal statement of the RANSAC paradigm, is used as the basis for determining that an n subset of P has been found that implies a sufficiently large consensus set to permit the algorithm to terminate. Thus, t must be chosen large enough to satisfy two purposes: that the correct model has been found for the data, and that a sufficient number of mutually consistent points have been found to satisfy the needs of the final smoothing procedure (which computes improved estimates for the model parameters).

To ensure against the possibility of the final consensus set being compatible with an incorrect model, and assuming that y is the probability that any given data point is within the error tolerance of an incorrect model, we would like y^{t-n} to be very small. While there is no general way of precisely determining y, it is certainly reasonable to assume that it is less than w (w is the *a priori* probability that a given data point is within the error tolerance of the correct model). Assuming $y < 0.5$, a value of $t - n$ equal to 5 will provide a better than 95 percent probability that compatibility with an incorrect model will not occur.

To satisfy the needs of the final smoothing procedure, the particular procedure to be employed must be specified. If least-squares smoothing is to be used, there are many situations where formal methods can be invoked

to determine the number of points required to produce a desired precision [10].

D. Example

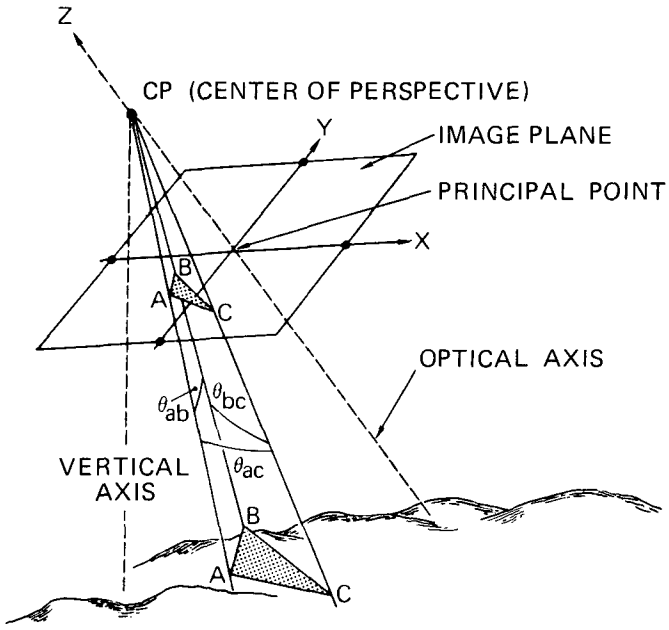
Let us apply RANSAC to the example described in Figure 1. A value of w (the probability that any selected data point is within the error tolerance of the model) equal to 0.85 is consistent with the data, and a tolerance (to establish datum/model compatibility) of 0.8 units was supplied as part of the problem statement. The RANSAC-supplied model will be accepted without external smoothing of the final consensus set; thus, we would like to obtain a consensus set that contains all seven data points. Since one of these points is a gross error, it is obvious that we will not find a consensus set of the desired size, and so we will terminate with the largest set we are able to find. The theory presented earlier indicates that if we take two data points at a time, compute the line through them and measure the deviations of the remaining points from this line, we should expect to find a suitable consensus set within two or three trials; however, because of the limited amount of data, we might be willing to try all 21 combinations to find the largest consensus set. In either case, we easily find the consensus set containing the six valid data points and the line that they imply.

III. The Location Determination Problem (LDP)

A basic problem in image analysis is establishing a correspondence between the elements of two representations of a given scene. One variation of this problem, especially important in cartography, is determining the location in space from which an image or photograph was obtained by recognizing a set of landmarks (control points) appearing in the image (this is variously called the problem of determining the elements of exterior camera orientation, or the camera calibration problem, or the image-to-database correspondence problem). It is routinely solved using a least-squares technique [11, 8] with a human operator interactively establishing the association between image points and the three-dimensional coordinates of the corresponding control points. However, in a fully automated system, where the correspondences must be based on the decisions of marginally competent feature detectors, least squares is often incapable of dealing with the gross errors that may result; this consideration, discussed at length in Sec. II, is illustrated for the LDP in an example presented in Sec. IV.

In this section a new solution to the LDP is presented based on the RANSAC paradigm, which is unique in its ability to tolerate gross errors in the input data. We will first examine the conditions under which a solution to the LDP is possible and describe new results concerning this question; we then present a complete description of the RANSAC-based algorithm, and finally, describe experimental results obtained through use of the algorithm.

Fig. 2. Geometry of the Location Determination Problem.



The LDP is formally defined as follows:

Given a set of m control points, whose 3-dimensional coordinates are known in some coordinate frame, and given an image in which some subset of the m control points is visible, determine the location (relative to the coordinate system of the control points) from which the image was obtained.

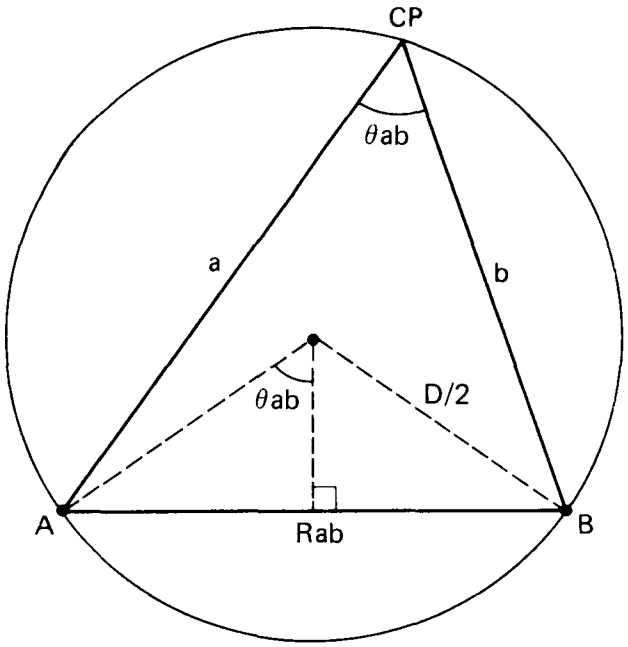
We will initially assume that we know the correspondences between n image points and control points; later we consider the situation in which some of these correspondences are invalid. We will also assume that both the principal point in the image plane (where the optical axis of the camera pierces the image plane) and the focal length (distance from the center of perspective to the principal point in the image plane) of the imaging system are known; thus (see Figure 2) we can easily compute the angle to any pair of control points from the center of perspective (CP). Finally, we assume that the camera resides outside and above a convex hull enclosing the control points.

We will later demonstrate (Appendix A) that if we can compute the lengths of the rays from the CP to three of the control points then we can directly solve for the location of the CP (and the orientation of the image plane if desired). Thus, an equivalent but mathematically more concise statement of the LDP is

Given the relative spatial locations of n control points, and given the angle to every pair of control points from an additional point called the Center of Perspective (CP), find the lengths of the line segments ("legs") joining the CP to each of the control points. We call this the "perspective-n-point" problem (PnP).

In order to apply the RANSAC paradigm, we wish to determine the smallest value of n for which it is possible to solve the PnP problem.

Fig. 3. Geometry of the P2P Problem.



$$\text{Sin } \theta ab = \frac{Rab/2}{D/2}$$

$$D = \frac{Rab}{\text{Sin } \theta ab}$$

A. Solution of the Perspective-*n*-Point Problem

The P1P problem ($n = 1$) provides no constraining information, and thus an infinity of solutions is possible. The P2P problem ($n = 2$), illustrated in Figure 3, also admits an infinity of solutions; the CP can reside anywhere on a circle of diameter Rab/sin(θab), rotated in space about the chord (line) joining the two control points A and B.

The P3P problem ($n = 3$) requires that we determine the lengths of the three legs of a tetrahedron, given the base dimensions and the face angles of the opposing trihedral angle (see Figure 4). The solution to this problem is implied by the three equations [A^*]:

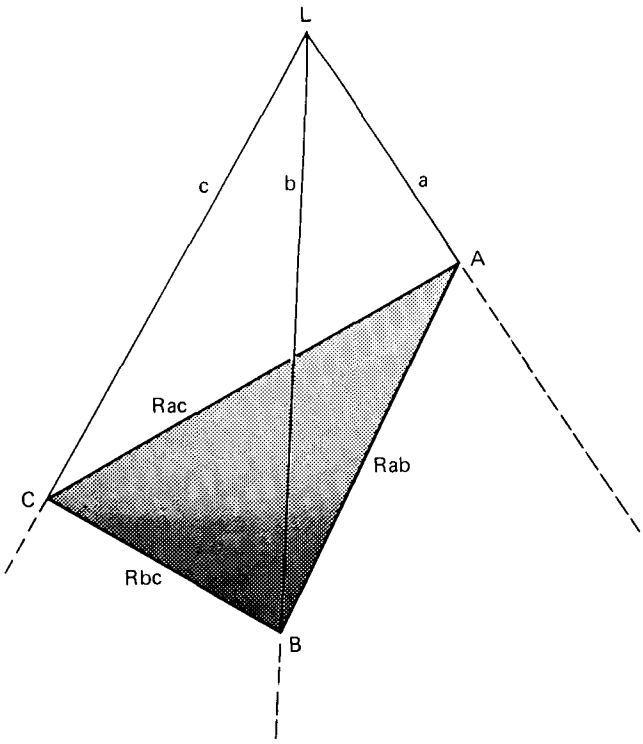
$$(Rab)^2 = a^2 + b^2 - 2^*a^*b^*[\cos(\theta ab)]$$

$$(Rac)^2 = a^2 + c^2 - 2^*a^*c^*\,[\cos(\theta ac)] \qquad [A^*]$$

$$(Rbc)^2 = b^2 + c^2 - 2^*b^*c^*\,[\cos(\theta bc)]$$

It is known that n independent polynomial equations, in n unknowns, can have no more solutions than the product of their respective degrees [2]. Thus, the system A^* can have a maximum of eight solutions. However, because every term in the system A^* is either a constant or of second degree, for every real positive solution there is a geometrically isomorphic negative solution. Thus, there are at most four positive solutions to A^*, and in Figure 5 we show an example demonstrating that the upper bound of four solutions is attainable.

Fig. 4. Geometry of the P3P Problem (L is the Center of Perspective).



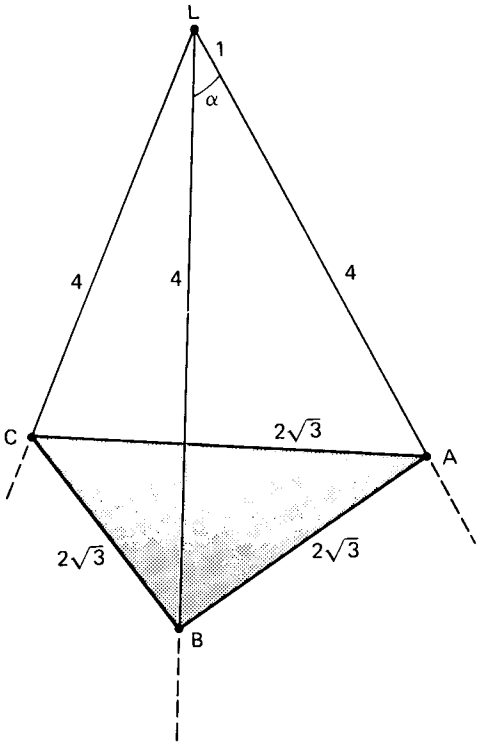
In Appendix A we derive an explicit algebraic solution for the system A^*. This is accomplished by reducing A^* to a biquadratic (quartic) polynomial in one unknown representing the ratio of two legs of the tetrahedron, and then directly solving this equation (we also present a very simple iterative method for obtaining the solutions from the given problem data).

For the case $n = 4$, when all four control points lie in a common plane (not containing the CP, and such that no more than two of the control points lie on any single line), we provide a technique in Appendix B that will always produce a unique solution. Surprisingly, when all four control points do not lie in the same plane, a unique solution cannot always be assured; for example, Figure 6 shows that at least two solutions are possible for the P4P problem with the control points in "general position."

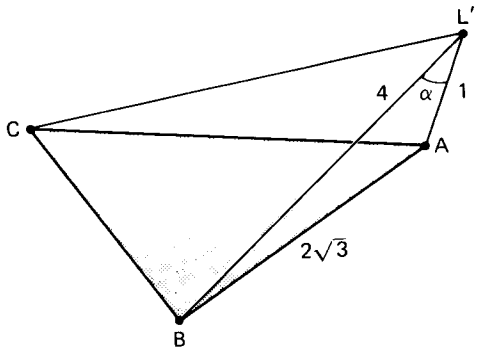
To solve for the location of the CP in the case of four nonplanar control points, we can use the algorithm presented in Appendix A on two distinct subsets of the control points taken three at a time; the solution(s) common to both subsets locate the CP to within the ambiguity inherent in the given information.

The approach used to construct the example shown in Figure 6 can be extended to any number of additional points. It is based on the principle depicted in Figure 3: If the CP and any number of control points lie on the same circle, then the angle between any pair of control points and the CP will be independent of the location on the circle of the CP (and hence the location of the CP cannot be determined). Thus, we are able to construct the example shown in Figure 7, in which five control points in general position imply two solutions to the P5P

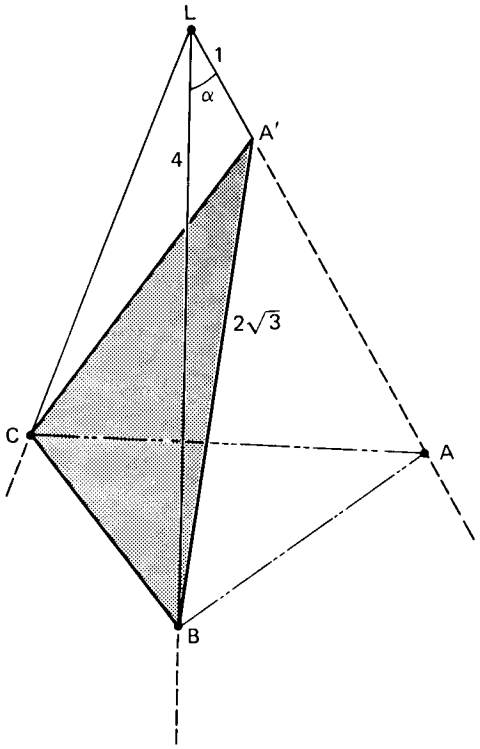
Fig. 5. An Example Showing Four Distinct Solutions to a P3P Problem.



(a)



(b)



(c)

Consider the tetrahedron in Figure 5(a). The base *ABC* is an equilateral triangle and the "legs" (i.e., *LA*, *LB*, and *LC*) are all equal. Therefore, the three face angles at *L* (i.e., <*ALB*, <*ALC*, and <*BLC*) are all equal. By the law of cosines we have:

$\text{Cos}(\alpha) = 5/8.$

This tetrahedron defines one solution to a P3P problem. A second solution is shown in Figure 5(b). It is obtained from the first by rotating *L* about *BC*. It is necessary to verify that the length of *L'A* can be 1, given the rigid triangle *ABC* and the angle alpha. From the law of cosines we have:

$(2*\sqrt{3})^2 = 4^2 + (L'A)^2 - 2*4*(L'A)*(5/8)$

which reduces to:

$(L'A - 1) * (L'A - 4) = 0.$

Therefore, *L'A* can be either 1 or 4. Figure 5(a) illustrates the *L'A* = 4 case and Figure 5(b) illustrates the *L'A* = 1 case.

Notice that repositioning the base triangle so that its vertices move to different locations on the legs is equivalent to repositioning *L*. Figure 5(c) shows the position of the base triangle that corresponds to the second solution.

Since the tetrahedron in Figure 5(a) is threefold rotationally symmetric, two more solutions can be obtained by rotating the triangle about *AB* and *AC*.

Fig. 6. An Example of a P4P Problem with Two Solutions.

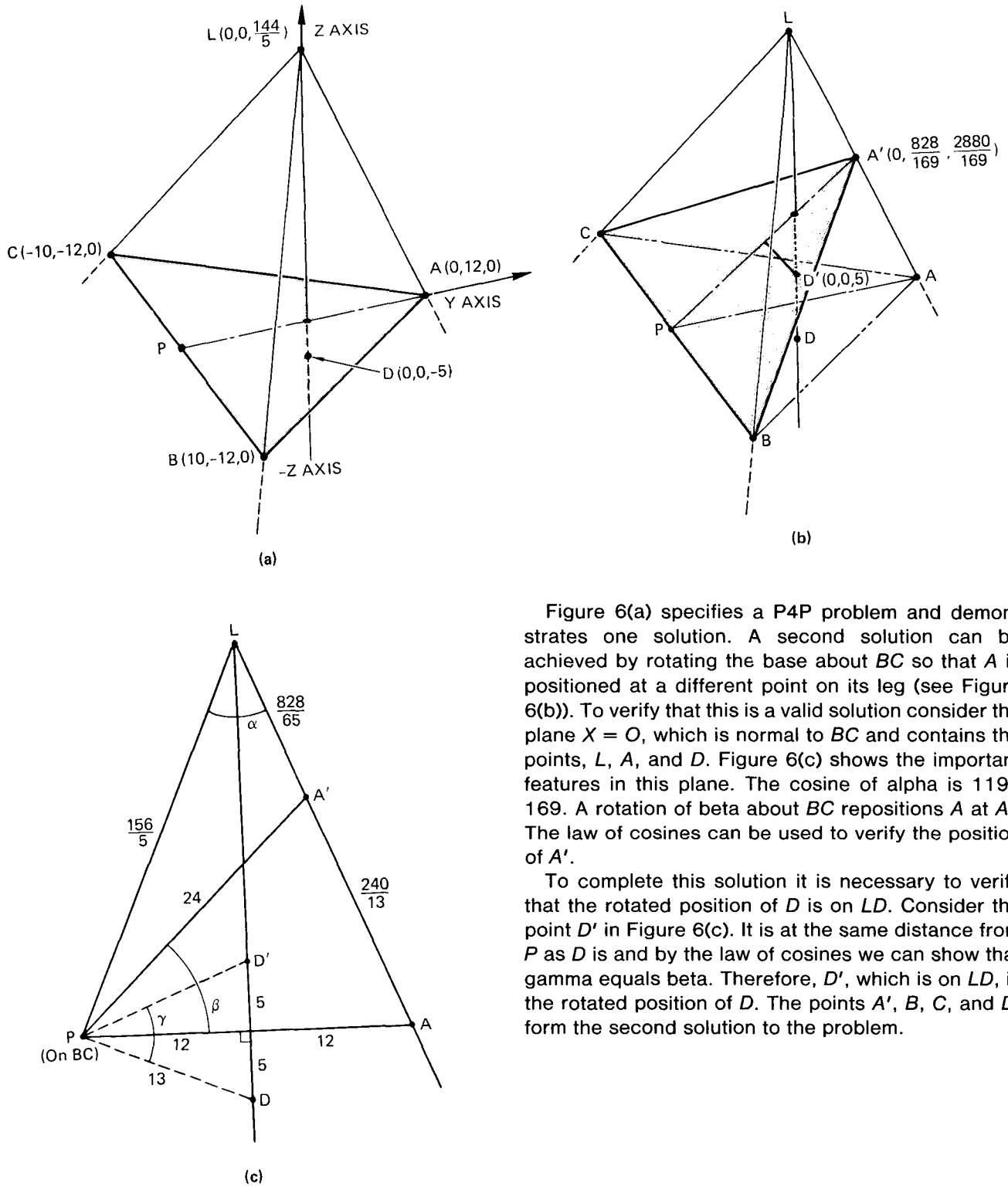


Figure 6(a) specifies a P4P problem and demonstrates one solution. A second solution can be achieved by rotating the base about *BC* so that *A* is positioned at a different point on its leg (see Figure 6(b)). To verify that this is a valid solution consider the plane $X = O$, which is normal to *BC* and contains the points, *L*, *A*, and *D*. Figure 6(c) shows the important features in this plane. The cosine of alpha is 119/169. A rotation of beta about *BC* repositions *A* at *A'*. The law of cosines can be used to verify the position of *A'*.

To complete this solution it is necessary to verify that the rotated position of *D* is on *LD*. Consider the point *D'* in Figure 6(c). It is at the same distance from *P* as *D* is and by the law of cosines we can show that gamma equals beta. Therefore, *D'*, which is on *LD*, is the rotated position of *D*. The points *A'*, *B*, *C*, and *D'* form the second solution to the problem.

problem. While the same technique will work for six or more control points, four or more of these points must now lie in the same plane and are thus no longer in general position.

To prove that six (or more) control points in general position will always produce a unique solution to the P6P problem, we note that for this case we can always solve for the 12 coefficients of the 3×4 matrix T that specifies the mapping (in homogeneous coordinates) from 3-space to 2-space; each of the six correspondences provides three new equations and introduces one additional unknown (the homogeneous coordinate scale factor). Thus, for six control points, we have 18 linear equations to solve for the 18 unknowns (actually, it can

be shown that, at most, 17 of the unknowns are independent). Given the transformation matrix T, we can construct an additional (synthetic) control point lying in a common plane with three of the given control points and compute its location in the image plane; the technique described in Appendix B can now be used to find a unique solution.

IV. Implementation Details and Experimental Results

A. The RANSAC/LD Algorithm

The RANSAC/LD algorithm accepts as input the following data:

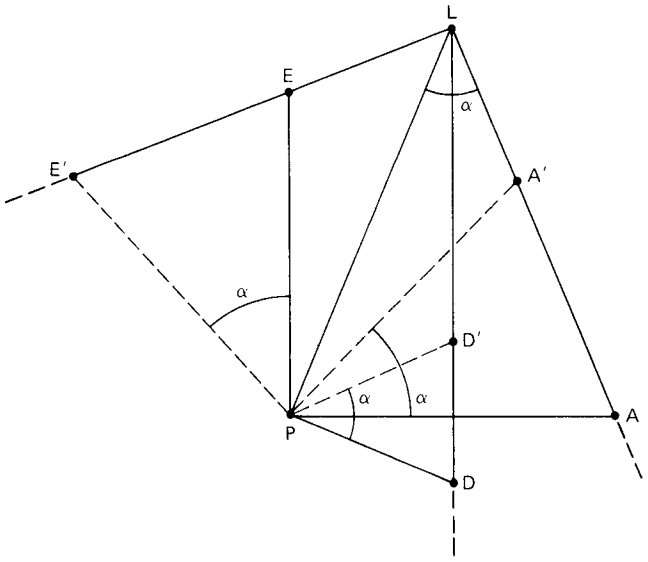
(1) A list L of m 6-tuples—each 6-tuple containing the 3-D spatial coordinates of a control point, its corresponding 2-D image plane coordinates, and an optional number giving the expected error (in pixels) of the given location in the image plane.

(2) The focal length of the imaging system and the image plane coordinates of the principal point.

(3) The probability $(1 - w)$ that a 6-tuple contains a gross mismatch.

(4) A "confidence" number G which is used to set the internal thresholds for acceptance of intermediate results contributing to a solution. A confidence number of one forces very conservative behavior on the algorithm; a confidence number of zero will call almost anything a valid solution.

Fig. 7. An Example of a P5P Problem with Two Solutions.



This example is the same as the P4P example described in Figure 6 except that a fifth control point, E, has been added. The initial position for E and its rotated position, E', are shown in Figure 7. The points E and E' were constructed to be the mirror images of A' and A about the line LP; therefore, a rotation of alpha about P repositions E at E'. One solution of the P5P problem is formed by points A, B, C, and D (shown in Figure 6(a)) plus point E. The second solution is formed by points A', B, C, D', and E'. Consequently there are two different positions of L such that all five points lie on their appropriate legs.

The RANSAC/LD algorithm produces as output the following information:

(1) The 3-D spatial coordinates of the lens center (i.e., the Center of Perspective), and an estimate of the corresponding error.

(2) The spatial orientation of the image plane.

The RANSAC/LD algorithm operates as follows:

(1) Three 6-tuples are selected from list L by a quasirandom method that ensures a reasonable spatial distribution for the corresponding control points. This initial selection is called S1.

(2) The CP (called CP1) corresponding to selection S1 is determined using the closed-form solution provided in Appendix A; multiple solutions are treated as if they were obtained from separate selections in the following steps.

(3) The error in the derived location of CP1 is estimated by perturbing the given image plane coordinates of the three selected control points (either by the amount specified in the 6-tuples or by a default value of one pixel), and recomputing the effect this would have on the location of the CP1.

(4) Given the error estimate for the CP1, we use the technique described in [1] to determine error ellipses (dimensions based upon the supplied confidence number) in the image plane for each of the control points specified in list L; if the associated image coordinates reside within the corresponding error ellipse, then the 6-tuple is appended to the consensus set S1/CP1.

(5) If the size of S1/CP1 equals or exceeds some threshold value t (nominally equal to a value between 7 and mw), then the consensus set S1/CP1 is supplied to a least-squares routine (see [1] or [7]) for final determination of the CP location and image plane orientation.[1] Otherwise, the above steps are repeated with a new random selection S2, S3, . . .

(6) If the number of iterations of the above steps exceeds $k = [\log(1 - G)]/[\log(1 - w^3)]$, then the largest consensus set found so far is used to compute the final solution (or we terminate in failure if this largest consensus set contains fewer than six members).

B. Experimental Results

To demonstrate the validity of our theoretical results, we performed three experiments. In the first experiment we found a specific LDP in which the common least-squares pruning heuristic failed, and showed that RANSAC successfully solved this problem. In the second experiment, we applied RANSAC to 50 synthetic problems in order to check the reliability of the approach over a wide range of parameter values. In the third experiment we used standard feature detection techniques to locate landmarks in an aerial image and then used RANSAC to determine the position and orientation of the camera.

C. A Location Determination Problem Example of a Least Squares Pruning Error

The LDP in this experiment was based upon 20 landmarks and their locations in an image. Five of the 20 correspondences were gross errors; that is, their given locations in the image were further than 10 pixels from their actual locations. The image locations for the good

[1] An alternative to least squares would be to average the parameters computed from random triples in the consensus set that fall within (say) the center 50 percent of the associated histogram.

correspondences were normally distributed about their actual locations with a standard deviation of one pixel.

The heuristic to prune gross errors was the following:

* Use all of the correspondences to instantiate a model.
* On the basis of that model, delete the correspondence that has the largest deviation from its predicted image location.
* Instantiate a new model without that correspondence.
* If the new model implies a normalized error for the deleted correspondence that is larger than three standard deviations, assume that it is a gross error, leave it out, and continue deleting correspondences. Otherwise, assume that it is a good correspondence and return the model that included it as the solution to the problem.

This heuristic successfully deleted two of the gross errors; but after deleting a third, it decided that the new model did not imply a significantly large error, so it returned a solution based upon 18 correspondences, three of which were gross errors. When RANSAC was applied to this problem, it located the correct solution on the second triple of selected points. The final consensus set contained all of the good correspondences and none of the gross errors.

D. 50 Synthetic Location Determination Problems

In this experiment RANSAC was applied to 50 synthetic LDPs. Each problem was based upon 30 landmark-to-image correspondences. A range of probabilities were used to determine the number of gross errors in the problems; the image location of a gross error was at least 10 pixels from its actual location. The location of a good correspondence was distributed about its actual location with a normal distribution having a standard deviation of one pixel. Two different camera positions were used—one looking straight down on the landmarks and one looking at them from an oblique angle. The RANSAC algorithm described earlier in this section was applied to these problems; however, the simple iterative technique described in Appendix A was used to locate solutions to the P3P problems in place of the closed form method also described in that appendix, and a second least-squares fit was used to extend the final consensus set (as suggested in Sec. II of this paper). Table I summarizes the results for ten typical problems (RANSAC successfully avoided including a gross error in its final consensus set in all of the problems); in five of these problems the probability of a good correspondence was 0.8, and in the other five problems, it was 0.6. The execution time for the current program is approximately 1 sec for each camera position considered.

E. A "Real" Location Determination Problem

Cross correlation was used to locate 25 landmarks in an aerial image taken from approximately 4,000 ft with a 6 in. lens. The image was digitized on a grid of 2,000 × 2,000 pixels, which implies a ground resolution of approximately 2 ft per pixel. Three gross errors were made by the correlation feature detector. When RANSAC was applied to this problem, it located a consensus set of 17 on the first triple selected and then extended that set to include all 22 good correspondences after the initial least-squares fit. The final standard deviations about the camera parameters were as follows:

X: 0.1 ft	Heading: 0.01°
Y: 6.4 ft	Pitch: 0.10°
Z: 2.1 ft	Roll: 0.12°

Table I. Typical Experimental Results Using RANSAC.

No. of Good Correspondences	No. of Correspondences in Final Consensus Set	No. of Triples Considered	No. of Camera Positions Considered
	$w = 0.8$		
22	19	6	10
23	23	1	3
19	19	2	3
25	25	1	2
24	23	3	8
	$w = 0.6$		
21	20	11	21
17	17	1	1
17	16	6	8
18	16	9	21
21	18	9	15

V. Concluding Comments

In this paper we have introduced a new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data. RANSAC is capable of interpreting/smoothing data containing a significant percentage of gross errors, and thus is ideally suited for applications in automated image analysis where interpretation is based on the data provided by error-prone feature detectors.

A major portion of this paper describes the application of RANSAC to the Location Determination Problem (LDP): Given an image depicting a set of landmarks with known locations, determine that point in space from which the image was obtained. Most of the results we presented concerning solution techniques and the geometry of the LDP problem are either new or not generally known. The current photogrammetric literature offers no analytic solution other than variants of least squares and the Church method for solving perspective-n-point problems. The Church method, which provides an iterative solution for the P3P problem [3, 11], is presented without any indication that more than one physically real solution is possible; there is certainly no indication that anyone realizes that physically real multiple solutions are possible for more than three control points in general position. (It should be noted that because the multiple solutions can be arbitrarily close together, even when an iterative technique is initialized to a value close to the correct solution there is no assurance that it will converge to the desired value.)

In the section on the LDP problem (and associated appendices) we have completely characterized the P3P problem and provided a closed-form solution. We have shown that multiple physically real solutions can exist for the P4P and P5P problems, but also demonstrated that a unique solution is assured when four of the control points reside on a common plane (solution techniques are provided for each of these cases). The issue of determining the maximum number of solutions possible for the P4P and P5P problems remains open, but we have shown that a unique solution exists for the P6P problem when the control points are in general position.

Appendix A. An Analytic Solution for the Perspective-3-Point Problem

The main body of this paper established that P3P problems can have as many as four solutions. In this appendix a closed form expression for obtaining these solutions is derived. Our approach involves three steps: (1) Find the lengths of the legs of the ("perspective") tetrahedron given the base (defined by the three control points) and the face angles of the opposing trihedral angle (the three angles to the three pairs of control points as viewed from the CP); (2) Locate the CP with respect to the 3-D reference frame in which the control points were originally specified; (3) Compute the orientation of the image plane with respect to the reference frame.

1. A Solution for the Perspective Tetrahedron (see Figure 4)

Given the lengths of the three sides of the base of a tetrahedron (Rab, Rac, Rbc), and given the corresponding face angles of the opposing trihedral angle (θab, θac, θbc), find the lengths of the three remaining sides of the tetrahedron (a, b, c).

A solution to the above problem can be obtained by simultaneously solving the system of equations:

$$(Rab)^2 = a^2 + b^2 - 2*a*b*\cos(\theta ab), \tag{A1}$$
$$(Rac)^2 = a^2 + c^2 - 2*a*c*\cos(\theta ac), \tag{A2}$$
$$(Rbc)^2 = b^2 + c^2 - 2*b*c*\cos(\theta bc). \tag{A3}$$

We now proceed as follows:

$$\text{Let } b = x*a \text{ and } c = y*a, \tag{A4}$$
$$(Rac)^2 = a^2 + (y^2)*(a^2) - 2*(a^2)*y*\cos(\theta ac), \tag{A5}$$
$$(Rab)^2 = a^2 + (x^2)*(a^2) - 2*(a^2)*x*\cos(\theta ab), \tag{A6}$$
$$(Rbc)^2 = (x^2)*(a^2) + (y^2)*(a^2) - 2*(a^2)*x*y*\cos(\theta bc). \tag{A7}$$

From Eqs. (A5) and (A7)

$$[(Rbc)^2]*[1 + (y^2) - 2*y*\cos(\theta ac)] = [(Rac)^2]*[(x^2) + (y^2) - 2*x*y*\cos(\theta bc)]. \tag{A8}$$

From Eqs. (A6) and (A7)

$$[(Rbc)^2]*[1 + (x^2) - 2*x*\cos(\theta ab)] = [(Rab)^2]*[(x^2) + (y^2) - 2*x*y*\cos(\theta bc)], \tag{A9}$$

$$\text{Let } \frac{(Rbc)^2}{(Rac)^2} = K1 \quad \text{and} \quad \frac{(Rbc)^2}{(Rab)^2} = K2. \tag{A10}$$

From Eqs. (A8) and (A9)

$$0 = (y^2)*[1 - K1] + 2*y*[K1*\cos(\theta ac) - x*\cos(\theta bc)] + [(x^2) - K1]. \tag{A11}$$

From Eqs. (A9) and (A10)

$$0 = (y^2) + 2*y*[-x*\cos(\theta bc)] + [(x^2)*(1 - K2) + 2*x*K2*\cos(\theta ab) - K2]. \tag{A12}$$

Equations (A11) and (A12) have the form:

$$0 = m*(y^2) + p*y + q, \tag{A13}$$
$$0 = m'*(y^2) + p'*y + q'. \tag{A14}$$

Multiplying Eqs. (A13) and (A14) by m' and m, respectively, and subtracting,

$$0 = [p*m' - p'*m]*y + [m'*q - m*q']. \tag{A15}$$

Multiplying Eqs. (A13) and (A14) by q' and q, respectively, substracting, and dividing by y,

$$0 = [m'*q - m*q']*(y^2) + [p'*q - p*q']*y,$$
$$0 = [m'*q - m*q']*y + [p'*q - p*q']. \tag{A16}$$

Assuming $m'*q \neq m*q'$,

$$[(x^2) - K1] \neq [(x^2)*(1 - K1)*(1 - K2) + 2*x*K2*(1 - K1)*\cos(\theta ab) - (1 - K1)*K2],$$

then Eqs. (A15) and (A16) are equivalent to Eqs. (A13) and (A14). We now multiply Eqs. (A15) by $(m'*q - m*q')$, and Eq. (A16) by $(p*m' - p'*m)$, and subtract to obtain

$$0 = (m'*q - m*q')^2 - [p*m' - p'*m]*[p'*q - p*q']. \tag{A17}$$

Expanding Eq. (A17) and grouping terms we obtain a biquadratic (quartic) polynomial in x:

$$0 = G4*(x^4) + G3*(x^3) + G2*(x^2) + G1*(x) + GO, \tag{A18}$$

where

$$G4 = (K1*K2 - K1 - K2)^2 - 4*K1*K2*[\cos(\theta bc)^2], \tag{A19}$$

$$G3 = 4*[K1*K2 - K1 - K2]*K2*(1 - K1)*\cos(\theta ab) + 4*K1*\cos(\theta bc)*[(K1*K2 + K2 - K1)*\cos(\theta ac) + 2*K2*\cos(\theta ab)*\cos(\theta bc)], \tag{A20}$$

$$G2 = [2*K2*(1 - K1)\cos(\theta ab)]^2 + 2*[K1*K2 + K1 - K2]*[K1*K2 - K1 - K2] + 4*K1*[(K1 - K2)*(\cos(\theta bc)^2) + (1 - K2)*K1*(\cos(\theta ac)^2) - 2*K2*(1 + K1)*\cos(\theta ab)*\cos(\theta ac)*\cos(\theta bc)], \tag{A21}$$

$$G1 = 4*(K1*K2 + K1 - K2)*K2*(1 - K1)*\cos(\theta ab) + 4*K1*[(K1*K2 - K1$$

$$+ K2)^* \cos(\theta ac)^* \cos(\theta bc) \quad \text{(A22)}$$
$$+ 2^*K1^*K2^* \cos(\theta ab)^* (\cos(\theta ac)^2)],$$

$$G0 = (K1^*K2 + K1 - K2)^2 - 4^*(K1^2)^*K2^* (\cos(\theta ac)^2). \quad \text{(A23)}$$

Roots of Eq. (A18) can be found in closed form [5], or by iterative techniques [4]. For each positive real root of Eq. (A18), we determine a single positive real value for each of the sides a and b. From Eq. (A6) we have

$$a = \frac{Rab}{SQRT[(x^2) - 2^*x^* \cos(\theta ab) + 1]}, \quad \text{(A24)}$$

and from Eq. (A4) we obtain

$$b = a^*x. \quad \text{(A25)}$$

If $m'^*q \neq m^*q'$, then from Eq. (A16) we have

$$y = \frac{p'^*q - p^*q'}{m^*q' - m'^*q}. \quad \text{(A26)}$$

If $m'^*q = m^*q'$, then Eq. (A26) is undefined and we obtain two values of y from Eq. (A5):

$$y = \cos(\theta ac) \pm SQRT\left[(\cos(\theta ac))^2 + \frac{(Rac)^2 - (a^2)}{(a^2)}\right]. \quad \text{(A27)}$$

For each real positive value of y, we obtain a value of c from Eq. (A4):

$$c = y^*a \quad \text{(A28)}$$

When values of y are obtained from Eq. (A5) rather than Eq. (A26), the resulting solutions can be invalid; they must be shown to satisfy Eq. (A3) before they are accepted.

It should be noted that because each root of Eq. (A18) can conceivably lead to two distinct solutions, the existence of the biquadratic does not by itself imply a maximum of four solutions to the P3P problem; some additional argument, such as the one given in the main body of this paper, is necessary to establish the upper bound of four solutions.

2. Example

For the perspective tetrahedron shown in Figure 5, we have the following parameters:

$$Rab = Rac = Rbc = 2^*SQRT(3),$$
$$\cos(\theta ab) = \cos(\theta ac) = \cos(\theta bc) = \frac{(a^2) + (b^2) - (Rab)^2}{2^*a^*b} = \frac{20}{32}.$$

Substituting these values into Eqs. (A19) through (A23), we obtain the coefficients of the biquadratic defined in Eq. (A18):

[−0.5625, 3.515625, −5.90625, 3.515625, −0.5625]

The roots of the above equation are

[1, 1, 4, 0.25]

For each root

Root	a	b	y	c
1	4	4	1	4
1	4	4	0.25	1
4	1	4	4	4
0.25	4	1	1	4

3. An Iterative Solution for the Perspective Tetrahedron (see Figure 8)

A simple way to locate solutions to P3P problems, which is sometimes an adequate substitute for the more involved procedure described in the preceding subsection, is to slide one vertex of the control-point triangle down its leg of the tetrahedron and look for positions of the triangle in which the other two vertices lie on their respective legs. If vertex A is at a distance a from L (L is the center of perspective), the lengths of the sides Rab and Rac restrict the triangle to four possible positions. Given the angle between legs LA and LB, compute the distance of point A from the line LB and then compute points $B1$ and $B2$ on LB that are the proper distance from A to insert a line segment of length Rab. Similarly, we compute at most two locations for C on its leg. Thus, given a position for A we have found at most four positions for a triangle that has one side of length Rab and one of length Rac. The lengths of the third sides (BC) of the four triangles vary nonlinearly as point A is moved down its leg. Solutions to the problem can be obtained by iteratively repositioning A to imply a third side of the required length.

4. Computing the 3-D Location of the Center of Perspective (see Figure 9)

Given the three-dimensional locations of the three control points of a perspective tetrahedron, and the lengths of the three legs, the 3-D location of the center of perspective can be computed as follows:

(1) Construct a plane $P1$ that is normal to AB and passes through the center of perspective, L. This plane can be constructed without knowing the position of L, which is what we are trying to compute. Consider the face of the tetrahedron that contains vertices A, B, and L. Knowing the lengths of sides LA, LB and AB, we can use the law of cosines to find the angle LAB, and then the projection QA of LA on AB. (Note that angle LQA is a right angle, and the point Q is that point on line AB that is closest to L). Construct a plane normal to AB passing through Q; this plane also passes through L.

(2) Similarly construct a plane $P2$ that is normal to AC and passes through L.

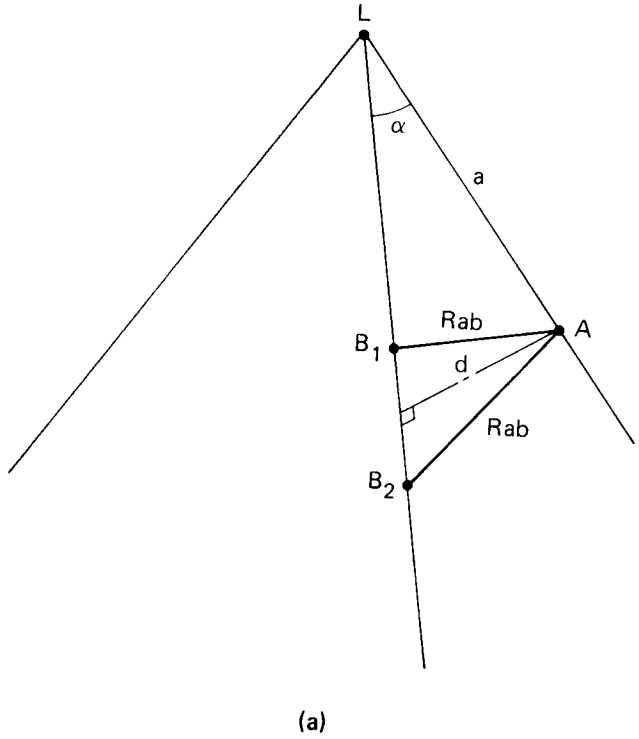
(3) Construct the plane $P3$ defined by the three points A, B, and C.

(4) Intersect planes $P1$, $P2$, and $P3$. By construction, the point of intersection R is the point on $P3$ that is closest to L.

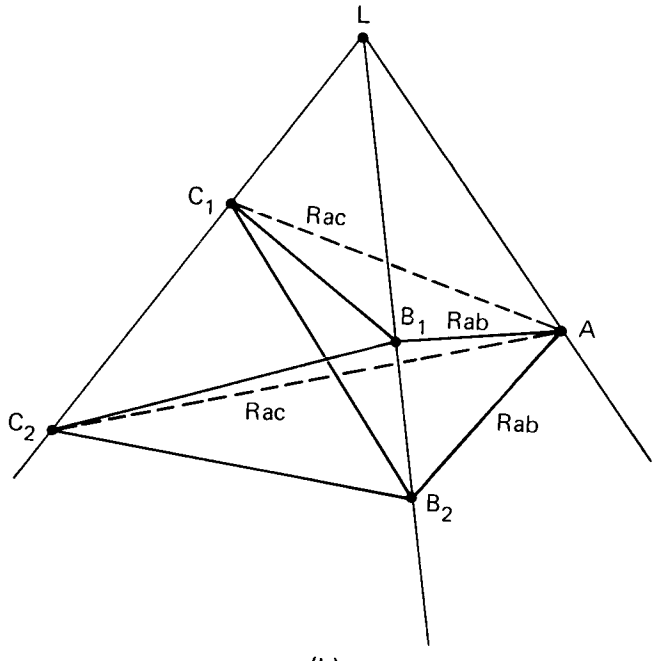
(5) Compute the length of the line AR and use that in conjunction with the length of LA to compute the length of the line RL, which is the distance of L from the plane $P3$.

(6) Compute the cross product of vectors AB and AC to form a vector perpendicular to $P3$. Then scale that vector by the length of RL and add it to R to get the 3-D location of the center of perspective L.

Fig. 8. Geometry for an Iterative Solution to the P3P Problem.



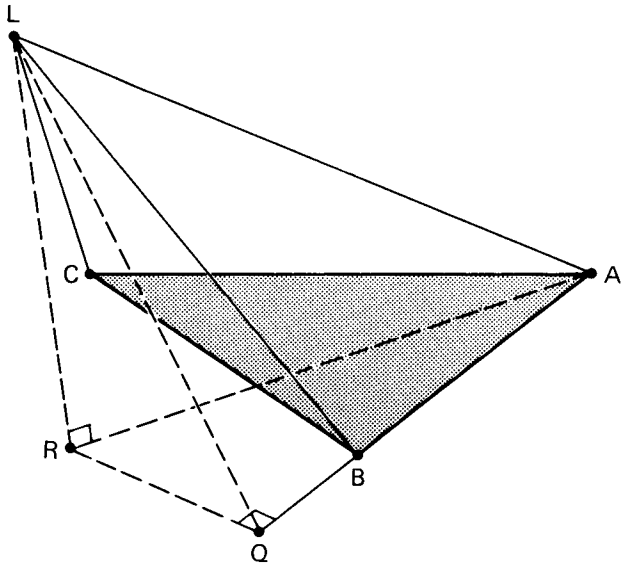
(a)



(b)

If the focal length of the camera and the principal point in the image plane are known, it is possible to compute the orientation of the image plane with respect to the world coordinate system; that is, the location of the origin and the orientation of the image plane coordinate system with respect to the 3-*D* reference frame. This can be done as follows:

Fig. 9 Computing the 3-*D* Location of the Center of the Perspective (*L*).



(1) Compute the 3-*D* reference frame coordinates of the center of perspective (as described above).

(2) Compute the 3-*D* coordinates of the image locations of the three control points: since we know the 3-*D* coordinates of the CP and the control points, we can compute the 3-*D* coordinates of the three rays between the CP and the control points. Knowing the focal length of the imaging system, we can compute, and subtract from each ray, the distance from the CP to the image plane along the ray.

(3) Compute the equation of the plane containing the image using the three points found in step (2). The normal to this plane, passing through the CP, gives us the origin of the image plane coordinate system (i.e., the 3-*D* location of the principal point), and the *Z* axis of this system.

(4) The orientation of the image plane about the *Z* axis can be obtained by computing the 3-*D* coordinates of a vector from the principal point to any one of the points found in step (2).

Appendix B. An Analytic Solution for the Perspective-4-Point Problem (with all control points lying in a common plane)

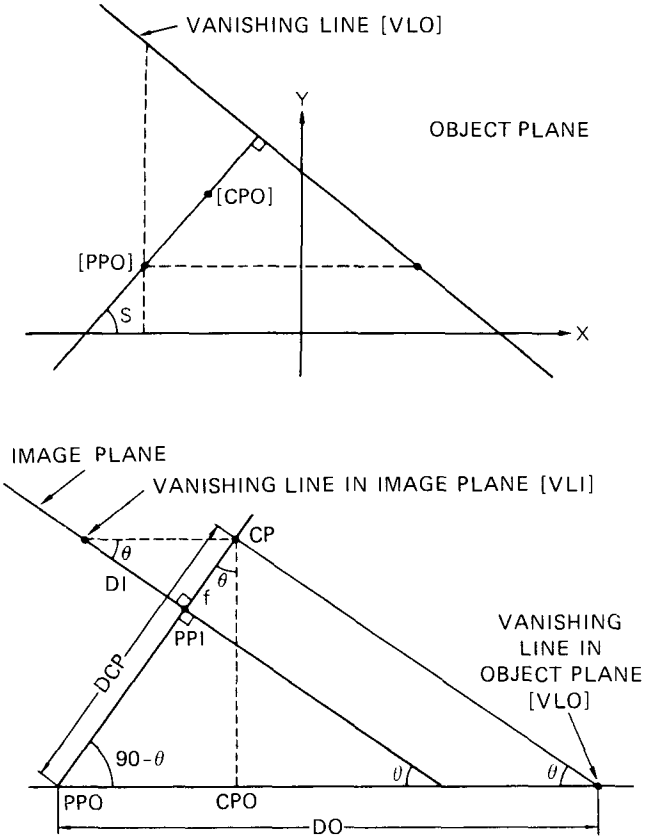
In this appendix an analytic technique is presented for obtaining a unique solution to the P4P problem when the four given control points all lie in a common plane.

1. Problem Statement (see Figure 10)

GIVEN: A correspondence between four points lying in a plane in 3-*D* space (called the object plane), and four points lying in a distinct plane (called the image plane); and given the distance between the center of perspective and the image plane (i.e., the focal length of the imaging system); and also given the principal point in the image plane (i.e., the location, in image plane coordinates, of the point at which the optical axis of the lense pierces the image plane).

FIND: the 3-*D* location of the center of perspective relative to the coordinate system of the object plane.

Fig. 10. Geometry of the P4P Problem (With all Control Points Lying in a Common Plane).



2. Notation

* Let the four given image points be labeled $\{Pi\}$, and the four corresponding object points $\{Qi\}$.
* We will assume that the 2-D image plane coordinate system has its origin at the principal point (PPI).
* We will assume that the object plane has the equation $Z = 0$ in the reference coordinate system. Standard techniques are available to transform from this coordinate system into a ground reference frame (e.g., see [6] or [9]).
* Homogeneous coordinates will be assumed [12].
* Primed symbols represent transposed structures.

3. Solution Procedure

(a) Compute the 3 × 3 collineation matrix T which maps points from object plane to image plane (a procedure for computing T is given later):

$$[Pi] = [T]*[Qi],$$

where

$$[Pi] = [ki*xi, ki*yi, ki]',$$
$$[Qi] = [Xi, Yi, 1]'. \quad \text{(B1)}$$

(b) The ideal line in the object plane, with coordinates [0, 0, 1]′ is mapped into the vanishing line in the image plane $[VLI]$ by the transformation:

$$[VLI] = [\text{inv}[T]]'*[0, 0, 1]'. \quad \text{(B2)}$$

(c) Determine the distance DI from the origin of the image plane (PPI) to the vanishing line $[VLI] = [a1, a2, a3]'$:

$$DI = \left| \frac{a3}{sqrt[(a1)^2 + (a2)^2]} \right|. \quad \text{(B3)}$$

(d) Solve for the dihedral (tilt) angle θ between the image and object planes:

$$\theta = \arctan(f/DI), \quad \text{(B4)}$$

where f = focal length.

(e) The ideal line in the image plane with coordinates [0, 0, 1]′ is mapped into the vanishing line in the object plane $[VLO]$ by the transform

$$[VLO] = [T]'*[0, 0, 1]'. \quad \text{(B5)}$$

(f) Compute the location of point $[PPO]$ in the object plane ($[PPO]$ is the point at which the optical axis of the lense pierces the object plane):

$$[PPO] = [\text{inv}[T]]'*[0, 0, 1]' \quad \text{(B6)}$$

(g) Compute the distance DO from $[PPO] = [c1, c2, c3]'$ to the vanishing line $[VLO] = [b1, b2, b3]'$ in the object plane:

$$DO = \left| \frac{b1*c1 + b2*c2 + b3*c3}{c3*sqrt[(b1)^2 + (b2)^2]} \right|. \quad \text{(B7)}$$

(h) Solve for the "pan" angle \$ as the angle between the normal to $[VLO] = [b1, b2, b3]'$ and the X axis in the object plane:

$$\$ = \arctan(-b2/b1). \quad \text{(B8)}$$

(i) Determine $XSGN$ and $YSGN$: If a line (parallel to the X axis in the object plane) through $[PPO]$ intersects $[VLO]$ to the right of $[PPO]$, then XSGN = 1. Otherwise $XSGN = -1$. Thus,

$$\text{if } \frac{b1*c1 + b2*c2 + b3*c3}{b1*c3} < 0,$$

then $XSGN = 1$, otherwise $XSGN = -1$. (B9)

Similarly,

$$\text{if } \frac{b1*c1 + b2*c2 + b3*c3}{b2*c3} < 0$$

then $YSGN = 1$, otherwise $YSGN = -1$. (B10)

(j) Solve for the location of the CP in the object plane coordinate system:

$$DCP = DO* \sin(\theta) \quad \text{(B11)}$$
$$XCP = XSGN*abs[DCP* \sin(\theta)* \cos(\$)] + c1/c3 \quad \text{(B12)}$$
$$YCP = YSGN*abs[DCP* \sin(\theta)* \sin(\$)] + c2/c3 \quad \text{(B13)}$$
$$ZCP = DCP* \cos(\theta) \quad \text{(B14)}$$

Note: If $[VLI]$, as determined in (b), has the coordinates [0, 0, k], then the image and object planes are parallel ($\theta = 0$). Rather than continuing with the above procedure, we now solve for the desired

information using similar triangles and Euclidean geometry.

4. Computing the Collineation Matrix T

Let

$$[Q] = \begin{Vmatrix} X1 & Y1 & 1 \\ X2 & Y2 & 1 \\ X3 & Y3 & 1 \end{Vmatrix} = [[Q1]', [Q2]', [Q3]'],$$

$$[P] = \begin{Vmatrix} x1 & y1 & 1 \\ x2 & y2 & 1 \\ x3 & y3 & 1 \end{Vmatrix} = [[P1]', [P2]', [P3]'],$$

$$[Q4] = [X4, Y4, 1]',$$

$$[P4] = [x4, y4, 1]',$$

$$[V] = [\text{inv}[P]]' * [P4] = [v1, v2, v3]',$$

$$[R] = [\text{inv}[Q]]' * [Q4] = [r1, r2, r3]',$$

$$w1 = \frac{v1}{r1} * \frac{r3}{v3},$$

$$w2 = \frac{v2}{r2} * \frac{r3}{v3},$$

$$[w] = \begin{Vmatrix} w1 & 0 & 0 \\ 0 & w2 & 0 \\ 0 & 0 & 1 \end{Vmatrix}.$$

Then,

$$[T]' = [\text{inv}[Q]]' * [W]' * [P]$$

such that

$$[Pi] = ki * [xi, yi, 1] = [T] * [Qi].$$

5. Example

Given:

f = 0.3048 m (12 in.)

$P1 = (-0.071263,\ 0.029665)$ $\quad Q1 = (-30,\ 80)$

$P2 = (-0.053033,\ -0.006379)$ $\quad Q2 = (-100,\ -20)$

$P3 = (-0.014063,\ 0.061579)$ $\quad Q3 = (140,\ 50)$

$P4 = (0.080120,\ -0.030305)$ $\quad Q4 = (-40,\ -240)$

(a)

$$[T]' = \begin{Vmatrix} 0.000212 & 0.000236 & 0.000925 \\ -0.000368 & 0.000137 & 0.000534 \\ -0.025404 & 0.021650 & 0.843879 \end{Vmatrix}$$

$$[\text{inv}[T]]' = \begin{Vmatrix} 1117.14 & -2038.86 & 0.0 \\ 3371.56 & 2302.22 & -5.14991 \\ -51.0636 & -120.442 & 1.31713 \end{Vmatrix}$$

(b) $[VLI] = [0, -5.14991, 1.31713]'$

(c) $DI = 0.255758$

(d) $\theta = 0.872665$ rad (50°)

(e) $[VLO] = [0.000925, 0.000534, 0.843880]'$

(f) $[PPO] = [-51.0636, -120.442, 1.31713]'$

(g) $DO = 711.196$

(h) $\$ = -0.523599$ rad (−30°)

(i) $XSGN = -1$

$YSGN = -1$

(j) $DCP = 544.8081$

$XCP = -400.202$

$YCP = -300.117$

$ZCP = 350.196$

Received 4/80; revised 1/81; accepted 1/81

References

1. Bolles, R.C., Quam, L.H., Fischler, M.A., and Wolf, H.C. The SRI road expert: Image to database correspondence. In Proc. Image Understanding Workshop, Pittsburgh, Pennsylvania, Nov., 1978.
2. Chrystal, G. *Textbook of Algebra* (Vol 1). Chelsea, New York, New York 1964, p. 415.
3. Church, E. Revised geometry of the aerial photograph. *Bull. Aerial Photogrammetry.* 15, 1945, Syracuse University.
4. Conte, S.D. *Elementary Numerical Analysis.* McGraw Hill, New York, 1965.
5. Dehn, E. *Algebraic Equations.* Dover, New York, 1960.
6. Duda, R.O., and Hart, P.E. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, 1973.
7. Gennery, D.B. Least-squares stereo-camera calibration. Stanford Artificial Intelligence Project Internal Memo, Stanford, CA 1975.
8. Keller, M. and Tewinkel, G.C. Space resection in photogrammetry. ESSA Tech. Rept C&GS 32, 1966, U.S. Coast and Geodetic Survey.
9. Rogers, D.P. and Adams, J.A. *Mathematical Elements for Computer Graphics.* McGraw Hill, New York, 1976.
10. Sorensen, H.W. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum* (July 1970), 63–68.
11. Wolf, P.R. *Elements of Photogrammetry.* McGraw Hill, New York, 1974.
12. Wylie, C.R. Jr. *Introduction to Projective Geometry.* McGraw-Hill, New York, 1970.