An Algorithm for Finding Repeated Solutions to the General Perspective Three-Point Pose Problem

Michael Q. Rieck

Received: date / Accepted: date

Abstract In the Perspective 3-Point Pose Problem (P3P), the transformation that converts the triple of (unknown) camera-to-control-point distances, into the triple of (known) angle cosines between the projection lines, is generally locally invertible. However, this fails to be the case when the camera's focal point (center of perspective) is on the danger cylinder. This situation corresponds to a double solution to P3P, and presents extra difficulties in solving P3P.

An extensive analysis of the danger cylinder setup leads to the introduction of a special rational function that proves to be quite useful in solving P3P in the danger cylinder case. This involves some rather long algebraic expressions that are best manipulated using mathematical software. Ultimately, some fairly simple formulas emerge that serve as a basis for an algorithm, called the *Double Solution Algorithm (DSA)*. Experimental results comparing DSA with Grunert's quartic polynomial method demonstrate that DSA often has substantially greater accuracy. This is particularly so when the camera is relatively far from the control points, even if it is not very close to the danger cylinder.

Keywords P3P, perspective, pose, danger cylinder, tracking

The author is grateful to the referees for several helpful suggestions, including a request for further data collection and analysis, resulting in better method comparisons.

Michael Rieck
Drake University,
Des Moines, IA, USA
Tel.: +515-271-3795
Fax: +515-271-2055
E-mail: mrieck@drake.edu

1 Introduction

After nearly two centuries, the Perspective 3-Point Pose Problem (P3P) continues to be of research interest, partly because of emerging applications, and partly because of its intrinsic and interesting mathematical complexity. While still of use in photogrammetry, its applications have widened into areas of digital imaging, to address a variety of practical problems. These include robotic control and navigation [7], [12], as well as six-degree-of-freedom tracking for virtual/augmented reality [1], [11].

P3P essentially amounts to determining the distances from a camera's focal point, also called the "center of perspective (CoP)," to three known "reference points," also called "control points." These control points are actual points in space whose images in the camera's image plane (or photograph) can be identified. The above goal is achieved by performing computations based on image measurements, intrinsic camera properties and the known physical distances between the control points.

Naturally, when tracking a camera, it is preferable to use more control points. Even with only four control points, it is easier to determine the distances, and often to do so with greater accuracy. However, it can easily happen that all but three control points move outside the view of the camera. (For example, this has often been the author's own experience tracking a Wii remote control device using light emitting diodes for control points.) When this limitation occurs, it is desirable to make the best possible effort to continue tracking correctly, using only three control points, until additional control points can again be captured in the image.

A situation which tends to cause particular difficulty when tracking with only three control points, is when the CoP is on or near the circular cylinder that contains the control points, and whose axis is perpendicular to the plane containing these points. This cylinder has come to be known as the "danger cylinder." The difficulty is caused by accumulated roundoff error, as discussed in this article and also in many of the cited articles.

Since the introduction and initial solution of the problem in [5], various efforts have been made by researchers to find other approaches to solving P3P, and to better understand the nature of the underlying system of equations. Some of the mid-twentieth century work, motivated largely by aerial reconnaissance issues, can be found in [8], [9], [14] and [16]. An extensive survey of the state of P3P as of 1994 can be found in [6]. Recent studies have classified solutions [3], [4], [15], [19], [20], produced new algorithms [2], [17], and considered generalizations of P3P [10]. A recent reexamination of the danger cylinder can be found in [21].

[13] undertakes the same goal as the present article, namely to effectively handle the danger cylinder case. However, its scope is limited to the special case in which the separation distances between the control points are equal. This greatly simplifies the problem, and the analysis and algorithm there is quite different than the analysis and algorithm in the present paper.

While the principal purpose of this article is to introduce an algorithm for more precisely solving P3P when near or on the danger cylinder, it contains a considerable development of some interesting mathematics underpinning this algorithm. This involves the introduction of a useful rational function called the "ratio function," which can be readily computed from a knowledge of the measured data, but which can also be expressed in terms of unknown quantities whose discovery leads immediately to a solution to P3P, in the danger cylinder case.

The next subsection introduces some basic mathematics of P3P. This preliminary analysis generalizes the notation and observations presented in the introduction of [13]. This is followed by a subsection that discusses the detection of repeated solutions to P3P, and the computational difficulties presented by such solutions. In Section 2, an algorithm called the "Double Solution Algorithm (DSA)" is introduced to overcome these difficulties. This section also presents the results of experiments that compare this algorithm against the quartic polynomial approach of Grunert. Section 3 lays out in detail the more elaborate mathematical analysis that underpins DSA.

1.1 Problem Statement and Basic Analysis

The P3P problem assumes that the cosines c_1, c_2, c_3 of the angles at the CoP subtended by the rays to the control points are known. These cosines are straightforward to calculate from the photograph (or digital image) and intrinsic camera properties. Following a common convention, c_1 denotes the cosine of the angle between the rays to the second and third control points, while c_2 denotes the cosine of the angle between the rays to the first and third control points, and c_3 denotes the cosine of the angle between the rays to the first and second control points. The distances d_1, d_2, d_3 between the control points are also presumed to be known. The subscripting convention here is similar to the subscripting of c_1, c_2, c_3.

The distances r_1, r_2, r_3 from the CoP to the control points are initially unknown, and need to be deduced. Using the Law of Cosines, the essential goal is to solve the following system of three quadratic equations:

$$\begin{cases} r_1^2 + r_2^2 - 2\,c_3\,r_1\,r_2 \;=\; d_3^2 \\ r_2^2 + r_3^2 - 2\,c_1\,r_2\,r_3 \;=\; d_1^2 \\ r_3^2 + r_1^2 - 2\,c_2\,r_3\,r_1 \;=\; d_2^2. \end{cases} \tag{1}$$

It will be convenient to set $R_j = r_j^2$ and $D_j = d_j^2$ for $j = 1, 2, 3$, and to sometimes regard (1) as a system of equations in the unknowns R_1, R_2 and R_3. Grunert [5] and others (see [6, Section 3]) discovered ways to solve this system exactly, by reducing it to a single quartic (*i.e.* fourth degree) polynomial in a single variable. However, the resulting quartic equation inevitably has coefficients that are complicated rational combinations of c_1, c_2, c_3, d_1, d_2 and d_3.

Nowadays, using mathematical manipulation software, it is relatively easy to reproduce Grunert's efforts by eliminating any two of R_1, R_2, R_3, and thus arrive at a quartic in the remaining R_j:

$$\mathcal{A}\,R_j^4 + \mathcal{B}_j\,R_j^3 + \mathcal{C}_j\,R_j^2 + \mathcal{D}_j\,R_j + \mathcal{E}_j \;=0 \quad (j = 1, 2, 3). \tag{2}$$

The coefficients here depend on c_1, c_2, c_3, D_1, D_2 and D_3. The leading coefficient $\mathcal{A}$, as well as the equation's discriminant $\varDelta$, turn out (surprisingly) to be independent of j. Specifically, $\mathcal{A} = 16\,T^2$ and

$$\begin{aligned} \varDelta \;=\; & 16777216\;T^2\;S\;\cdot \\ & \left[(D_1{-}D_2{+}D_3)c_1^2 + (D_1{-}D_2{-}D_3)c_2^2 + 2(D_2{-}D_1)\tau\right]^2 \cdot \\ & \left[(D_2{-}D_3{+}D_1)c_2^2 + (D_2{-}D_3{-}D_1)c_3^2 + 2(D_3{-}D_2)\tau\right]^2 \cdot \\ & \left[(D_3{-}D_1{+}D_2)c_3^2 + (D_3{-}D_1{-}D_2)c_1^2 + 2(D_1{-}D_3)\tau\right]^2, \end{aligned} \tag{3}$$

where $\tau = c_1c_2c_3$, $T = 1 + 2\tau - c_1^2 - c_2^2 - c_3^2$, and where S is a complicated polynomial in c_1, c_2, c_3, D_1, D_2 and D_3. The quartic equation (2) can then be solved exactly using classical methods. [1]

Eliminating say r_3 from (1) is straightforward actually. Subtract the bottom two equations, and thus produce an equation that is only linear in r_3. Solve this for r_3. Use this to substitute for r_3 in the second equation, to obtain an equation involving r_1 and r_2 only. The first equation in (1) is another such equation. In this way, (1) reduces to a system of two equations in two unknowns. Reducing this further to a single equation in a single variable is considerably harder. The complete reductions for Grunert's method, as well as a few related methods, are detailed in [6].

For fixed d_1, d_2, d_3, consider the rational transformation $(r_1, r_2, r_3) \mapsto (c_1, c_2, c_3)$ via

$$c_1 = \frac{r_2^2 + r_3^2 - d_1^2}{2r_2r_3}, \quad c_2 = \frac{r_3^2 + r_1^2 - d_2^2}{2r_3r_1},$$
$$c_3 = \frac{r_1^2 + r_2^2 - d_3^2}{2r_1r_2}, \tag{4}$$

obtained by solving (1) for the c_j. When this is used to express the c_j in terms of the r_j and d_j, the polynomial S becomes a rational function of the R_j and D_j. Specifically,

$$S = \frac{\Omega^2 H}{256\, R_1^4\, R_2^4\, R_3^4}\,, \tag{5}$$

where

$$\begin{aligned}\Omega &= D_1D_2D_3 + (D_1 + D_2 - D_3)R_1R_2 \\ &+ (D_2 + D_3 - D_1)R_2R_3 + (D_3 + D_1 - D_2)R_3R_1 \\ &- D_1R_1^2 - D_2R_2^2 - D_3R_3^2,\end{aligned} \tag{6}$$

and where H is a complicated polynomial in D_1, D_2, D_3, R_1, R_2 and R_3. Moreover, the Jacobian determinant of the transformation (4) is

$$J = \begin{vmatrix} \frac{\partial c_1}{\partial r_1} & \frac{\partial c_1}{\partial r_2} & \frac{\partial c_1}{\partial r_3} \\ \frac{\partial c_2}{\partial r_1} & \frac{\partial c_2}{\partial r_2} & \frac{\partial c_2}{\partial r_3} \\ \frac{\partial c_3}{\partial r_1} & \frac{\partial c_3}{\partial r_2} & \frac{\partial c_3}{\partial r_3} \end{vmatrix} = \frac{\Omega}{4\, r_1^3 r_2^3 r_3^3}\,. \tag{7}$$

[1] The computations required here and elsewhere in this article are quite tedious, and best checked using mathematical manipulation software, such as Mathematica® or Maple™. A Mathematica notebook is available from the author upon request.

1.2 Double Solutions

Repeated roots to the quartic polynomial in (2) occur if and only if $\Delta = 0$. But $\Delta = 0$ if and only if one of the factors in the factorization (3) is zero. It turns out, however, that a repeated solution to (1), meaning coalescing solutions having the same values of R_1, R_2 *and* R_3, occurs if and only if $S = 0$, which occurs if and only if there is a solution for which the camera's CoP is on the danger cylinder.

Now, $S = 0$ if and only if $\Omega = 0$ or $H = 0$. Again, $S = 0$ is a necessary and sufficient condition for a repeated solution to exist. $\Omega = 0$ means that the solution being considered in the formula for Ω is a repeated solution. It turns out that the condition $H = 0$ means that two *other* solutions to (1), for the same values of c_1, c_2 and c_3, coalesce to form a double solution.

In addition, by (7), the Jacobian J vanishes if and only if Ω does, and when this occurs, the transformation (4) is no longer locally invertible. This makes recovering the values of r_1, r_2 and r_3 more complicated, given the values of c_1, c_2, c_3, d_1, d_2 and d_3. Even when J is nonzero but is small, methods for finding r_1, r_2 and r_3 become quite prone to round-off errors.

If there is a reason to suspect that the CoP is on or sufficiently near the danger cylinder, then it might be reasoned that methods like Grunert's cannot be relied on to accurately find the solution. Instead, the algorithm in the next subsection (DSA) can be used to more accurately find a double solution or at least find a number very close to two nearly equal solutions.

It is important to have a practical procedure to flag when the CoP has come too close to the danger cylinder. It must be assumed that in using Grunert's method, some good procedure already exists for selecting the appropriate solution from among the several solutions yielded by this method. We must depend on Grunert (or a similar method) to provide fairly accurate values for r_1, r_2, r_3. The selection could be based on using a history of the tracking in order to assign a likelihood of being correct to each of these solutions.

Now, to decide if the camera has wandered too close to the danger cylinder, a check might be made to see if two highly likely solutions are sufficiently close together. A superior alternative to this test, according to experiments discussed in Subsection 2.2, is to focus on the quantity J, and detect when this is sufficiently small so as to warrant triggering the use of DSA. Actually, in the experiments in Section 2.2, a variation of this turned out to be more useful, namely, criterion (8).

2 The Double Solution Algorithm (DSA)

The algorithm to be presented, for finding a repeated solution to P3P, was motivated by the algorithm presented in [13], for the special case where the control points are the vertices of an equilateral triangle. Now though, we are considering a general setup for the perspective three-point pose problem, and the algorithm is quite different. However, both algorithms are based on detailed analyses of the transformation (4), restricted to the danger cylinder, and they share a number of aspects. At the risk of creating a little confusion, the author feels that the present algorithm should also be called the "Double Solution Algorithm," and when necessary, context can supply a distinction between this algorithm and the one in [13].

2.1 Algorithm Specification

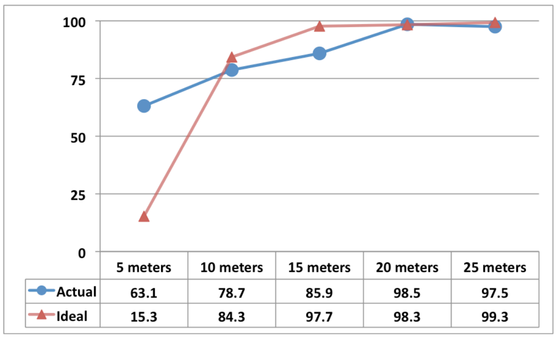
The present algorithm splits into two parts, the first of which, called "Preprocess," depends only on a knowledge of the separation distances between the control points. As long as these points remain fixed, this part of the algorithm only needs to be performed once.

The second part, called "Main", needs to be executed each time the cosines c_1, c_2, and c_3 are measured, in the case of a moving camera taking repeated measurements. It is assumed that Main has access to all the information produced by Preprocess.

Here now are the algorithm details:

Preprocess

1. Receive (d_1, d_2, d_3) as input.
2. With $D_1 = d_1^2, D_2 = d_2^2, D_3 = d_3^2$, compute $2r^2$ for the circumradius r, using (11).
3. Rescale by replacing (D_1, D_2, D_3) with (D_1, D_2, D_3) divide by $2r^2$.
4. Set $v_1 = 1-D_2$, $v_2 = 1-D_1$, $v_3 = 1$, $w_1 = \sqrt{1-v_1^2}$, $w_2 = \pm\sqrt{1-v_2^2}$, and $w_3 = 0$. The sign for w_2 is chosen to make $v_1v_2 + w_1w_2 = 1 - D_3$.
5. Set $t_1 = w_1/(1+v_1), t_2 = w_2/(1+v_2), t_3 = 0, s_1 = t_1 + t_2, s_2 = t_1t_2$ and $s_3 = 0$.
6. Compute $\kappa, \zeta, \bar{\zeta}, \lambda', \lambda''$ and λ''', using (21), (24) and (25).
7. Setting $\lambda = \lambda'$, λ'', λ''', successively, compute quintuples $(\mu', \nu', \gamma_1', \gamma_2', \gamma_3')$, $(\mu'', \nu'', \gamma_1'', \gamma_2'', \gamma_3'')$ and $(\mu''', \nu''', \gamma_1''', \gamma_2''', \gamma_3''')$, as the corresponding values for $(\mu, \nu, \gamma_1, \gamma_2, \gamma_3)$, using (22) and (23).
8. Compute β_0 as in Fact 3 in Section 3.
9. Return all the values computed here.



	5 meters	10 meters	15 meters	20 meters	25 meters
● Actual	63.1	78.7	85.9	98.5	97.5
▲ Ideal	15.3	84.3	97.7	98.3	99.3

Fig. 1 DSA usage percentages for two techniques

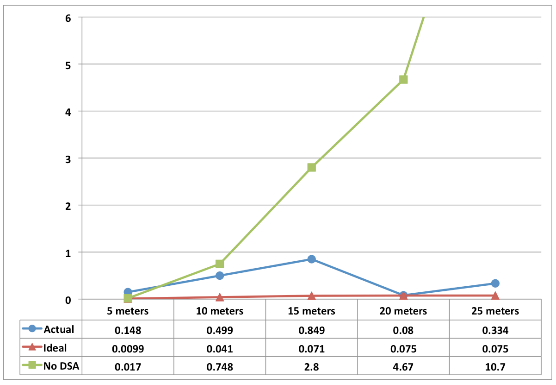
Main

1. Receive (c_1, c_2, c_3) as input.
2. Compute $c_1^2, c_2^2, c_3^2, \tau$ and T. (See (3).)
3. Successively compute $\rho(-\beta_0-\gamma_1-\gamma_2-\gamma_3, \beta_0, \gamma_1, \gamma_2, \gamma_3;\ t_1, t_2, t_3;\ t, u)$, using (20), for the three values of the triplet $(\gamma_1, \gamma_2, \gamma_3)$ produced in Step 7 of Preprocess. Call the resulting values ρ', ρ'' and ρ'''.
4. Consider the three pairs of quadratic equations in (27). For each combination of a choice from each pair, test to see it the resulting system of three quadratic equations has a common root, as discussed at the end of the Section 3. For the combination that produces a common root, set t to equal this root.
5. Using Facts 4 and 7 in Section 3, compute the appropriate $\rho(\cdots)$, using the known values for t_1, t_2, t_3 and t.
6. Solve the resulting quadratic equation for u.
7. Compute R_1, R_2 and R_3 using (16) and similar formulas.
8. Rescale by replacing (R_1, R_2, R_3) with (R_1, R_2, R_3) multiplied by $2r^2$.
9. Compute $r_1 = \sqrt{R_1}, r_2 = \sqrt{R_2}$ and $r_3 = \sqrt{R_3}$.
10. Return (r_1, r_2, r_3).

Step 4 in Main assumes the CoP is on the danger cylinder. However, when this is only approximately the case, then t needs to be set to an approximate common root instead, as discussed in Subsection 3.3. No root extractions are required in this step. In fact, root extractions only occur in Steps 6 and 9, and these are just square roots.

2.2 Experimental Results

Extensive simulations were conducted to compare DSA against Grunert's method, using single precision float-



	5 meters	10 meters	15 meters	20 meters	25 meters
● Actual	0.148	0.499	0.849	0.08	0.334
▲ Ideal	0.0099	0.041	0.071	0.075	0.075
■ No DSA	0.017	0.748	2.8	4.67	10.7

Fig. 2 Average error amounts for three techniques

ing point data. [2] Triangles were randomly generated and oriented, and placed so that the triangle's plane was at a prescribed distance from the CoP, using distances of 5, 10, 15, 20 and 25 meters. The vertices of a triangle served as control points during applications of Grunert's method and the DSA method.

While triangles of various shapes were checked, the figures presented here are based on data collected for triangles having one side randomly set between 4 and 6 meters, another side randomly set between 5 and 7 meters, and the remaining side randomly set between 6 and 8 meters. For each triangle, a random point on the danger cylinder was chosen, and the methods were tested with the CoP placed somewhere along a one-meter-long line segment, centered at the random point. For placement of the CoP, one hundred equally-spaced locations along the segment (extending to the ends) were each used in succession, and the results from these locations averaged. These averages were then averaged over the random triangles.

First Grunert's method was used. Based on the results, a criterion was applied to decide whether or not switching to the DSA method was warranted. Several possible criteria were explored. The results presented in the figures are based on the seemingly arbitrary criterion

$$|\Omega| < 0.2\,(R_1 + R_2 + R_3)^2. \tag{8}$$

This worked reasonably well for the triangles tested here, and for the wide range of distances between these and the CoP. No explanation for why this was a successful criterion is offered.

More specifically, the plots in the figures labeled "actual" represent data collected when this criterion was used to decide. The plot labeled "no DSA" is based on strictly using Grunert's method without potentially switching to DSA. The plots labeled "ideal" show what would happen if a perfect criterion was available for deciding whether or not to use DSA in favor of Grunert. In this case, both methods were always executed and compared, and the better result was always chosen.

As said, these experiments began by checking the method of Grunert, which can result in the need to select the best solution from up to four solutions to the system of equations (1). In practice, this remains a tricky issue, which can be addressed by using a tracking history to predict the likelihood of correctness for each of these solutions. This problem is not addressed in this paper, and the experiments performed were based on the assumption of always making this choice correctly.

Figure 1 indicates the percentage of times that the DSA was selected for use and its results used in place of Grunert's method. As said, the "actual" plot is based on using DSA if and only if (8) is satisfied. By contrast, the "ideal" plot shows the percentage of times that DSA *should* be used, because it actually produced better results.

Figure 2 records average error measurements. This measurement is the sum of squared distances between the three actual control points and their predicted positions in space (relative to the CoP), based on the method used to solve the system (1). When DSA was not used at all, this error grew very large as the CoP was located at increasing distances from the control points. Also, when the distance between the CoP and the plane containing the control points was 20 meters, the criterion (8) gave results that were extremely close to ideal. This did not continue at further distances, however.

Using the timing features of C++, it was found that DSA did not impose an unreasonable performance penalty. In the "ideal" experiments, where both the method of Grunert and DSA are always used, it was found that no more than a sixty percent performance penalty was experienced, as compared with using only Grunert's method. Of course the delay caused by an "actual" experiment would depend (linearly) on the percentage of times that DSA is actually used. The simple criterion test (8) imposes only an insignificant delay.

3 Mathematical Analysis

3.1 The Danger Cylinder and Rationalization

The analysis here begins by choosing a rectangular coordinate system such that the three control points are located at coordinates $(r\cos\theta_j, r\sin\theta_j, 0)$, for $j = 1, 2, 3$. Assuming the camera's CoP to be on the danger cylin-

[2] The source code for the C++ program used for data collection and analysis is available from the author upon request

der, its coordinates are $(r\cos\theta, r\sin\theta, z)$ for some θ and z. Now,

$$D_1 = d_1^2 = r^2(\cos\theta_2 - \cos\theta_3)^2 + r^2(\sin\theta_2 - \sin\theta_3)^2$$
$$= 2r^2\left[1 - \cos(\theta_2 - \theta_3)\right] = 4r^2 \sin^2\left(\frac{\theta_2 - \theta_3}{2}\right). \tag{9}$$

Similarly for D_2 and D_3. Likewise,

$$R_1 = r_1^2 = r^2(\cos\theta - \cos\theta_1)^2 + r^2(\sin\theta - \sin\theta_1)^2 + z^2$$
$$= 2r^2\left[1 - \cos(\theta - \theta_1)\right] + z^2 = 4r^2 \sin^2\left(\frac{\theta - \theta_1}{2}\right) + z^2, \tag{10}$$

and similarly for R_2 and R_3.

d_1, d_2, d_3 and r are of course dependent. In fact, a standard formula for the circumradius of a triangle tells us that

$$r^2 = \frac{D_1 D_2 D_3}{2D_1D_2 + 2D_2D_3 + 2D_3D_1 - D_1^2 - D_2^2 - D_3^2} \tag{11}$$

By rescaling, there is no harm in assuming that $r = 1/\sqrt{2}$, so that $2r^2 = 1$. This assumption will henceforth be imposed. If we set $u = z^2, v = \cos\theta$, $w = \sin\theta$, $v_j = \cos\theta_j$, and $w_j = \sin\theta_j$ $(j = 1, 2, 3)$, then the D_j and R_j can be expressed as polynomials in these, as follows:

$$D_1 = 1 - v_2v_3 - w_2w_3, \quad D_2 = 1 - v_3v_1 - w_3w_1,$$
$$D_3 = 1 - v_1v_2 - w_1w_2, \quad \text{and} \tag{12}$$
$$R_1 = 1 - v_1v - w_1w + u, \quad R_2 = 1 - v_2v - w_2w + u,$$
$$R_3 = 1 - v_3v - w_3w + u. \tag{13}$$

However, v and w are not independent, and of course are related by $v^2 + w^2 = 1$. Similarly for v_j and w_j $(j = 1, 2, 3)$. These dependencies complicate the manipulation of expressions involving these quantities when using mathematical manipulation software to explore long combinations. Fortunately, the situation can be easily remedied, by leveraging a standard rational parameterization of the unit circle. Specifically, new symbols t_1, t_2, t_3 and t can be introduced, as independent variables/parameters, and we can set

$$v = \frac{1 - t^2}{1 + t^2}, \quad w = \frac{2t}{1 + t^2}, \tag{14}$$
$$v_j = \frac{1 - t_j^2}{1 + t_j^2}, \quad w_j = \frac{2t_j}{1 + t_j^2} \qquad (j = 1, 2, 3).$$

Notice that $t = w/(1 + v)$ and $t_j = w_j/(1 + v_j)$, which can be used to set the parameters t_1, t_2 and t_3, from a knowledge of the separation distances d_1, d_2 and d_3. There is some flexibility though in the choice of the coordinate system. We may assume, for instance, that the third control point is on the positive half of the x-axis, whence $v_3 = 1, w_3 = 0, v_1 = 1 - D_2$ and $v_2 = 1 - D_1$. The DSA algorithm makes this assumption. In this section though, the points will be kept general, to exhibit the symmetry in t_1, t_2 and t_3. The following formulas are straightforward to deduce:

$$D_1 = \frac{2(t_2 - t_3)^2}{(1 + t_2^2)(1 + t_3^2)}, \tag{15}$$

$$R_1 = \frac{(1 + t_1^2)(1 + t^2)u + 2(t - t_1)^2}{(1 + t_1^2)(1 + t^2)} \tag{16}$$

By symmetry, there are similar formulas for D_2, D_3, R_2 and R_3. From these, we obtain

$$R_2 + R_3 - D_1 = \tag{17}$$
$$\frac{2(1 + t_2^2)(1 + t_3^2)(1 + t^2)u + 4(1 + t_2t_3)(t - t_2)(t - t_3)}{(1 + t_2^2)(1 + t_3^2)(1 + t^2)},$$

and similar formulas for $R_3 + R_1 - D_2$ and $R_1 + R_2 - D_3$. Since $c_1^2 = (R_2 + R_3 - D_1)^2 / (4R_2R_3)$, etc., we obtain

$$c_1^2 = \tag{18}$$
$$\begin{aligned}&\left[(1 + t_2^2)(1 + t_3^2)(1 + t^2)u + 2(1 + t_2t_3)(t - t_2)(t - t_3)\right]^2 \\ &/ \ \{ \ (1 + t_2^2)\left[\ (1 + t_2^2)(1 + t^2)\,u + 2(t - t_2)^2 \ \right] \\ &\quad \cdot (1 + t_3^2)\left[\ (1 + t_3^2)(1 + t^2)\,u + 2(t - t_3)^2 \ \right] \ \},\end{aligned}$$

and so forth. Now, $\tau = (R_2 + R_3 - D_1)(R_3 + R_1 - D_2)$ $(R_1 + R_2 - D_3) / (8R_1R_2R_3)$, so we also have

$$\tau = \tag{19}$$
$$\begin{aligned}&\left[(1 + t_2^2)(1 + t_3^2)(1 + t^2)\,u + 2(1 + t_2t_3)(t - t_2)(t - t_3)\right] \\ &\cdot \left[(1 + t_3^2)(1 + t_1^2)(1 + t^2)u + 2(1 + t_3t_1)(t - t_3)(t - t_1)\right] \\ &\cdot \left[(1 + t_1^2)(1 + t_2^2)(1 + t^2)u + 2(1 + t_1t_2)(t - t_1)(t - t_2)\right] \\ &/ \ \{ \ (1 + t_1^2)\left[(1 + t_1^2)(1 + t^2)\,u + 2(t - t_1)^2\right] \\ &\quad \cdot (1 + t_2^2)\left[(1 + t_2^2)(1 + t^2)\,u + 2(t - t_2)^2\right] \\ &\quad \cdot (1 + t_3^2)\left[(1 + t_3^2)(1 + t^2)\,u + 2(t - t_3)^2\right] \ \}.\end{aligned}$$

The formulas encountered henceforth grow considerably in complexity. Without mathematical manipulation software, the final results seem to be too tedious to derive "by hand." Therefore, *no proofs are offered for the correctness of the subsequent formulas.* However, the discussion should hopefully provides a clear sense of how the formulas can be reproduced. In addition, thorough and separate testings of DSA, using Mathematica and C++, strongly supports the claim for the correctness of the formulas, and for the methodology presented here.

3.2 The "Ratio Function" ρ

Going forward, we will focus on certain rational combinations of c_1^2, c_2^2, c_3^2 and τ $(= c_1c_2c_3)$. As just seen, all four of these quantities in turn have rational expressions in terms of t_1, t_2, t_3, t and u, so that we will ultimately be concerned with rational functions of these. It will be helpful to introduce the elementary symmetric polynomials $s_1 = t_1+t_2+t_3,\ s_2 = t_1t_2+t_2t_3+t_3t_1,\ s_3 = t_1t_2t_3$. Recall too that $T = 1 + 2\tau - c_1^2 - c_2^2 - c_3^2$.

The following rational function was motivated by the analysis developed in [13] for the equilateral-triangle case. It has proven to be very useful for the general case, though this is much more complicated. Define the *ratio function*

$$\rho\,(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3;\ t_1, t_2, t_3;\ t, u) \quad = \quad \frac{\alpha + \beta\tau + \gamma_1 c_1^2 + \gamma_2 c_2^2 + \gamma_3 c_3^2}{T}\,. \tag{20}$$

For fixed $\alpha, \beta, \gamma_1, \gamma_2$ and γ_3, this quantity can be viewed alternatively as a rational function of t_1, t_2, t_3, t and u, or as a rational function of τ, c_1^2, c_2^2 and c_3^2. Due to the latter expression, for specific choices of the parameters $\alpha, \beta, \gamma_1, \gamma_2$ and γ_3, its value becomes immediately available once the cosines c_1, c_2 and c_3 are measured. In fact, we shall soon restrict to special values of the parameters which being known a priori (based on an a priori knowledge of t_1, t_2, t_3) combine with a knowledge of the measured cosines, so as to enable the determination of the values of the unknowns t and u, and thence v, w, R_1, R_2 and R_3. This provides the foundation of the Double Solution Algorithm that was presented in the previous section.

We begin with some interesting observations concerning the ratio function, which were discovered using mathematical manipulation software.

Facts concerning ρ

1. $\rho\,(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3;\ t_1, t_2, t_3;\ t, u)$, as a function of the variables t and u, has a numerator containing non-zero terms corresponding only to these monomials: 1, t, t^2, t^3, t^4, t^5, t^6, u, ut, ut^2, ut^3, ut^4, ut^5, ut^6, u^2, u^2t, u^2t^2, u^2t^3, u^2t^4, u^2t^5, u^2t^6, u^3, u^3t^2, u^3t^4, u^3t^6.

2. When $\alpha + \beta + \gamma_1 + \gamma_2 + \gamma_3 = 0$, the terms in the numerator corresponding to these monomials vanish as well: u^2t, u^2t^3, u^2t^5, u^3, u^3t^2, u^3t^4, u^3t^6.

3. When additionally $\beta = [\,(1+t_1^2)(t_2-t_3)^2\gamma_1 + (1+t_2^2)(t_3-t_1)^2\gamma_2 + (1+t_3^2)(t_1-t_2)^2\gamma_3\,]\ /\ [\,3s_1s_3 - s_1^2 - s_2^2 + 3s_2\,]$, the only non-zero terms in the numerator correspond to these monomials: u, ut, ut^2, ut^3, ut^4, ut^5, ut^6. (Call the value of β in this case β_0.)

4. When $\alpha = -(1+t_1t_2)(1+t_2t_3)(1+t_3t_1)$, $\beta = (1+t_1^2)(1+t_2^2)(1+t_3^2)$, and $\gamma_1 = \gamma_2 = \gamma_3 = 0$, the only non-zero terms in the numerator correspond to these monomials: u, ut, ut^2, ut^3, ut^4, u^2, u^2t, u^2t^2, u^2t^3, u^2t^4, u^3, u^3t^2, u^3t^4.

5. The denominator of $\rho\,(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3;\ t_1, t_2, t_3;\ t, u)$ is $4(t_1-t_2)^2(t_2-t_3)^2(t_3-t_1)^2\ (1+t^2)^3\,u$.

6. In the case described in Fact 3, $(1+t^2)\,u$ is a factor of the numerator, which thus cancels with a corresponding factor in the denominator. This eliminates u, leaving a quartic in t divided by a constant times $(1-t^2)^2$. (The constant depends on t_1, t_2, t_3.)

7. In the case described in Fact 4, u cancels out between the numerator and denominator. If values are now supplied for t_1, t_2, t_3 and t, then the result is a quadratic polynomial in u.

3.3 Discovering t and u

Fact 7 can be of use in determining u once t has been determined – just solve a quadratic equation. Additionally, Fact 6 suggests a strategy for discovering the value of t. The difficulty however is that even though the value of $\rho(\cdots)$ is knowable, being easily determined from c_1, c_2 and c_3, when this is used together with Fact 6, we still *seem* to need to solve a quartic equation, in order to obtain t. *Not so!* This is because we have liberty in choosing the values for $\gamma_1, \gamma_2, \gamma_3$, and it turns out that these can be selected so as to make the numerator become the square of a quadratic. In fact, this can be accomplished *in three distinct ways.*

Consider the equation whose left side is the numerator of $\rho\,(-\beta_0 - \gamma_1 - \gamma_2 - \gamma_3, \beta_0, \gamma_1, \gamma_2, \gamma_3;\ t_1, t_2, t_3;\ t, u)$ (expressed in terms of t_1, t_2, t_3, t, u), and whose right side is $(t^2 + \mu t + \nu)^2$, for unknowns μ and ν. Equating the coefficients of t, we obtain five equations in the unknowns $\gamma_1, \gamma_2, \gamma_3, \mu$ and ν. Using elimination, it is possible to reduce this to a cubic equation in μ, with positive discriminant $(108[s_1 - s_3]^4[1 + s_1^2 - 2s_2 + s_2^2 - 2s_1s_3 + s_3^2]^2)$. Consequently, there are three distinct real-valued choices for μ, though when limited to only extracting radicals, they must be expressed as combinations of complex numbers (*casus irreducibilis*, cf. [18, pp.189-190]).

Since the five equations are linear in $\gamma_1, \gamma_2, \gamma_3$, row reduction can be used to write each of these in terms of μ and ν. Moreover, it turns out that after eliminating $\gamma_1, \gamma_2, \gamma_3$, it is possible to produce an equation involving μ and ν that only involves ν linearly. Letting

$$\kappa = (t_1 - t_2)(t_2 - t_3)(t_3 - t_1)(3s_1s_3 - s_2^2 - s_1^2 + 3s_2), \tag{21}$$

and noting that the denominator of $\rho(\cdots)$ is $\kappa\,(1+t^2)^2$, one discovers that

$$\nu \;=\; \frac{(s_2-1)[\,(s_3-s_1)\mu^2-3(s_2-1)\mu-2(s_3-s_1)\,]}{(s_3-s_1)[\,(s_3-s_1)\mu-2(s_2-1)\,]}, \tag{22}$$

$$\gamma_1 = \frac{\begin{array}{c}-2t_1(1+t_1^2)(t_2+t_3)^2\ \mu\nu \\ +\ t_1(s_2-1)[\,s_3-t_1+(t_1^2+2)(t_2+t_3)\,]\ \nu^2 \\ +\ (t_2t_3-t_3t_1-t_1t_2-1)(s_2-1)\end{array}}{(t_1^2+1)(s_2-1)^2\ \kappa},$$

$$\gamma_2 = \frac{\begin{array}{c}-2t_2(1+t_2^2)(t_3+t_1)^2\ \mu\nu \\ +\ t_2(s_2-1)[\,s_3-t_2+(t_2^2+2)(t_3+t_1)\,]\ \nu^2 \\ +\ (t_3t_1-t_1t_2-t_2t_3-1)(s_2-1)\end{array}}{(t_2^2+1)(s_2-1)^2\ \kappa},$$

$$\gamma_3 = \frac{\begin{array}{c}-2t_3(1+t_3^2)(t_1+t_2)^2\ \mu\nu \\ +\ t_3(s_2-1)[\,s_3-t_3+(t_3^2+2)(t_1+t_2)\,]\ \nu^2 \\ +\ (t_1t_2-t_2t_3-t_3t_1-1)(s_2-1)\end{array}}{(t_3^3+1)(s_2-1)^2\ \kappa}.$$

The cubic equation for μ can be put in the form

$$\lambda^3+p\lambda+q=0\,, \tag{23}$$

where

$$\lambda=\mu+\frac{2(s_2-1)}{s_1-s_3},\quad q=\frac{2(s_2-1)p}{3(s_1-s_3)}\quad \text{and}$$

$$p=\frac{-3(1+s_1^2-2s_2+s_2^2-2s_1s_3+s_3^2)}{(s_1-s_3)^2}.$$

Setting

$$\begin{aligned}\zeta &= [-(1+it_1)(1+it_2)(1+it_3)]^{1/3} \\ &= [s_2-1+(s_3-s_1)\,i\,]^{1/3}\end{aligned}$$

and (24)

$$\begin{aligned}\bar\zeta &= [-(1-it_1)(1-it_2)(1-it_3)]^{1/3} \\ &= [s_2-1+(s_1-s_3)\,i\,]^{1/3}\end{aligned}$$

the roots of this cubic are found to be

$$\lambda'=\frac{\zeta\,\bar\zeta\,(\zeta+\bar\zeta)}{s_1-s_3},\ \lambda''=\frac{\zeta\,\bar\zeta\,[(1-\sqrt{-3})\,\zeta+(1+\sqrt{-3})\,\bar\zeta]}{2(s_3-s_1)}$$

$$\text{and }\lambda'''=\frac{\zeta\,\bar\zeta\,[(1+\sqrt{-3})\,\zeta+(1-\sqrt{-3})\,\bar\zeta]}{2(s_3-s_1)}. \tag{25}$$

For each of the three values for λ, corresponding values of μ, ν, γ_1, γ_2 and γ_3 are then uniquely determined. Let us call the resulting quintuples of values (μ', ν', γ_1', γ_2', γ_3'), (μ'', ν'', γ_1'', γ_2'', γ_3'') and (μ''', ν''', γ_1''', γ_2''', γ_3'''). It is important to notice that all of these values can be computed with *only* a knowledge of d_1, d_2 and d_3. These computations can precede the acquisition of the data (c_1, c_2, c_3) from the camera image, and only need to be performed once, as long as the control points remain fixed.

Now, once the values of c_1, c_2 and c_3 are acquired, three different $\rho(\cdots)$ values can be computed, using the three different settings for the triple $(\gamma_1,\gamma_2,\gamma_3)$ obtained above, and setting $\beta=\beta_0$ and $\alpha=-\beta_0-\gamma_1-\gamma_2-\gamma_3$. Calling the resulting values ρ', ρ'' and ρ''', three equations emerge:

$$\begin{aligned}(t^2+\mu't+\nu')^2 &= \kappa\,\rho'\,(1+t^2)^2, \\ (t^2+\mu''t+\nu'')^2 &= \kappa\,\rho''\,(1+t^2)^2, \\ (t^2+\mu'''t+\nu''')^2 &= \kappa\,\rho'''\,(1+t^2)^2.\end{aligned} \tag{26}$$

Thus,

$$\begin{aligned}t^2+\mu't+\nu' &= \pm\sqrt{\kappa\rho'}\,(1+t^2), \\ t^2+\mu''t+\nu'' &= \pm\sqrt{\kappa\rho''}\,(1+t^2), \\ t^2+\mu'''t+\nu''' &= \pm\sqrt{\kappa\rho'''}\,(1+t^2).\end{aligned} \tag{27}$$

There are three pairs of quadratic equations here. Some selection of an equation from each pair must result in a system of three equations with a simultaneous solution. Given any two quadratic equations, it is easy to test whether or not they have a common root, using standard linear elimination methods. Specifically, at^2+bt+c and dt^2+et+f (with $bd\neq ae$ and $af\neq cd$) have a common root if and only if $(af-cd)^2=(bd-ae)(ce-bf)$, in which case the common root is $(af-cd)\,/\,(bd-ae)$. Moreover, the quantity $|(af-cd)^2-(bd-ae)(ce-bf)|$ can be used as a metric for checking how close two quadratics are to having a common root.

This provides a fast technique for deciding which three of the above six equations to select, and for determining their common (or nearly common) root. In this way, we can quickly obtain the correct value of t. From this, u can be quickly obtained as well, using Fact 7, as mentioned earlier. So R_1, R_2 and R_3 are now determined, by (16) and similar formulas.

4 Conclusion

A detailed algorithm (DSA) has been introduced for better handling the perspective three-point pose problem, when the camera is on or near the danger cylinder. Experimental results confirm that DSA can be dramatically more accurate than a certain traditional approaches, particularly at long distances. Simple criteria for reasonably deciding when and when not to use DSA have been seen to be possible. However, further exploration of this aspect of DSA is needed.

In analyzing the difficulties presented by the danger cylinder, and overcoming these, it was very useful to “rationalize” the problem (by introducing t_1, t_2, t_3

and t), and to focus on a particular parameterized rational function (the "ratio function"). This analysis ultimately suggested DSA, and demonstrates its correctness. However, it involved somewhat tedious derivations that were made using mathematical manipulation software. It would be highly desirable to discover a more intuitive way of arriving at the same (or better) results. Possibly this could result from a better geometric sense of the equations obtained here.

References

1. Chen, C-S., Hung, Y-P., Shih, S-W., Hsieh, C-C., Tang, C-Y., Yu, C-G. and Chang, Y-C.: Integrating virtual objects into real images for augmented reality. *VRST'98, ACM Symp. Virtual Reality Software and Techology*, 1-8 (1998)
2. DeMenthon, D., Davis,L. S.: Exact and approximate solutions of the perspective-three-point problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(11), 1100-1105 (1992)
3. Faugère, J-C., Moroz, G., Roullier, F., El Din, M. S.: Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities. *ISSAC'08, 21st ACM Int. Symp. Symbolic and Algebraic Computation*, 79-86 (2008)
4. Gao, X-S., Hou, X-R., Tang, J., Cheng, H-F.: Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8), 930-943 (2003)
5. Grunert, J. A.: Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1, 238-248 (1841)
6. Haralick, R. M., Lee, C-N., Ottenberg, K., Nölle, N.: Review and analysis of solutions of the three point perspective pose estimation Problem. *J. Computer Vision*, 13(3), 331-356 (1994)
7. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects: A survey. em Foundations and Trends in Computer Graphics and Vision, 1(1), 1-89 (2005)
8. Merritt, E. L.: Explicit three-point resection in space. *Photogrammetric Engineering*, 15(4), 649-655 (1949).
9. Müller, F. J.: Direkte (exakte) lösung des einfachen rückwärtsein-schneidens im raume. *Allegemaine Vermessungs-Nachrichten* (1925)
10. Nistér, D.: A minimal solution to the generalized 3-point pose problem. *J. Mathematical Imaging and Vision*, 27(1), 67-79 (2007)
11. Ohayon, S., Rivlin, E.: Robust 3D head tracking using camera pose estimation. *ICIP'06, Int. Conf. Image Processing*, 1063-1066 (2006)
12. Qingxuan, J., Ping, Z., Hanxu, S.: The study of positioning with high-precision by single camera based on P3P algorithm. *INDIN'06, IEEE Int. Conf. Industrial Informatics*, 1385-1388 (2006)
13. Rieck, M. Q.: Handling repeated solutions to the perspective three-point pose problem. *VISAPP '10, Int. Conf. Computer Vision Theory and Appl.* (to appear) (2010)
14. Smith, A. D. N.: The explicit solution of the single picture resolution problem, with a least squares adjustment to redundant control. *Photogrammetric Record*, 5(26), 113-122 (1965)
15. Tang, J., Liu, N.: The unique solution for P3P problem. *SIGAPP '09, ACM Symp. Applied Computing*, 1138-1139 (2009)
16. Thompson, E. H.: Space resection: failure cases. *Photogrammetric Record*, 5(27), 201-204 (1966)
17. Xiaoshan, G., Hangfei, C.: New algorithms for the perspective-three-point problem. *J. Comput. Sci. & Tech.*, 16(3), 194-207 (2001)
18. Van der Waerden, B. L.: *Algebra, v. 1.*, Frederick Unger Publishing, New York (1970)
19. Wolfe, W. J., Mathis, D., Sklair, C. W., Magee, M.: The perspecive view of three points In *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(1), 66-73 (1991)
20. Zhang, C-X., Hu, Z-Y.: A general sufficient condition of four positive solutions of the P3P problem. *J. Comput. Sci. & Technol.*, 20(6), 836-842 (2005)
21. Zhang, C-X., Hu, Z-Y. Why is the danger cylinder dangerous in the P3P problem? *Acta Automatica Sinica*, 32(4), 504-511 (2006)