

HANDLING REPEATED SOLUTIONS TO THE PERSPECTIVE THREE-POINT POSE PROBLEM

Michael Q. Rieck
Drake University, U.S.A.

Keywords: P3P, Pose, Photogrammetry, Danger cylinder, Discriminant, Jacobian, Repeated solution, Double solution.

Abstract: In the Perspective 3-Point Pose Problem (P3P), when the three reference points are equidistant from each other, this distance may be assumed to be one unit in length. A repeated solution to the problem then occurs when and only when $1 + R_1R_2 + R_2R_3 + R_3R_1 - R_1^2 - R_2^2 - R_3^2 = 0$, where R_1, R_2 and R_3 are the squared distances from the camera's focal point to the reference points. When the setup only approximately satisfies this equation, two nearly equal solutions can introduce substantial calculation errors. To better handle this circumstance, it may be preferable to behave as though the above equation holds precisely, and then invert a certain two-dimensional transformation to obtain the repeated solution. The inversion involves only a few basic arithmetic operations and square roots. This approach is more efficient, and more reliable, than the standard quartic equation approach to solving P3P, at least in this special case.

1 INTRODUCTION

The Perspective 3-Point Pose Problem (P3P), as introduced and solved by J. A. Grunert (1841), is essentially concerned with inferring the distances to three known reference points, seen in a photograph, from the camera that took the photograph. With this information, one can then determine the position and orientation of the camera. Amazingly, the problem is nearly as old as photography itself.

Traditionally its applications were restricted to areas of photogrammetry such as aerial reconnaissance. More recently though, it has been successfully applied in electronic digital imaging to address a variety of practical problems. These include robotic control and navigation, as in (Qingxuan, et al, 2006), as well as six-degree-of-freedom tracking for virtual/augmented reality and video game applications, as in (Chen, et al, 1998) and (Ohayon and Rivlin, 2006).

Advancements and refinements in the study of P3P were steadily made throughout the nineteenth and twentieth century, as for example (Merritt, 1949) and (Müller, 1925). An extensive survey of the state of P3P as of 1994 can be found in (Haralick, et al, 1994). Several recent studies have classified solutions, such as (Faugère, et al, 2008), (Gao, et al, 2003), (Wolfe, et al, 1991), (Zhang and Hu, 2005)

and (Zhang and Hu, 2006).

A simplified version of P3P assumes that the distances between the three reference points are equal. Attention will be limited in the following discussion to this situation, where in fact, the measurement units will be set so as to make this distance equal one. The P3P problem assumes that the cosines of the interior angles between pairs of lines-of-sight to the reference points are known, and that one wishes to determine the distances to these points. These cosines are straightforward to calculate from the photograph (or digital image) and intrinsic camera properties. Using the Law of Cosines, the underlying mathematical problem to be solved is therefore the determination of the unknown values of r_1, r_2, r_3 , based on the known values of c_1, c_2, c_3 , in the following system of quadratic equations:

$$\begin{cases} r_1^2 + r_2^2 - 2c_3r_1r_2 = 1 \\ r_2^2 + r_3^2 - 2c_1r_2r_3 = 1 \\ r_1^2 + r_3^2 - 2c_2r_3r_1 = 1. \end{cases} \quad (1)$$

It will be convenient to set $R_j = r_j^2$ ($j = 1, 2, 3$), and to sometimes regard (1) as a system of equations in R_1, R_2 and R_3 . As demonstrated by Grunert, it is possible to eliminate any two of the three unknowns, resulting in a single quartic (*i.e.* fourth degree) polynomial equation in the remaining R_j (for $j = 1, 2, 3$):

$$AR_j^4 + B_jR_j^3 + C_jR_j^2 + D_jR_j + E_j = 0. \quad (2)$$

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

$$\Delta = 16777216 (c_1^2 - c_2^2)^2 (c_2^2 - c_3^2)^2 (c_3^2 - c_1^2)^2 T^2 S, \quad (3)$$

where

$$T = 1 + 2\tau - \sigma, \quad S = \quad (4)$$

$$4(1 - \tau)^2 (1 + 8\tau) T - 3[3\chi - (1 + 2\tau)^2]^2,$$

$$\sigma = c_1^2 + c_2^2 + c_3^2, \quad \tau = c_1 c_2 c_3, \quad \chi = c_1^2 c_2^2 + c_2^2 c_3^2 + c_3^2 c_1^2.$$

The computations involved here are rather tedious, and best checked using mathematical manipulation software, such as Mathematica[®] or Maple[™].¹ The polynomial S appears to be irreducible. However, the story changes when the c_j are expressed in terms of the r_j , using the following rational transformation:

$$c_1 = \frac{r_2^2 + r_3^2 - 1}{2r_2 r_3}, \quad c_2 = \frac{r_3^2 + r_1^2 - 1}{2r_3 r_1}, \quad c_3 = \frac{r_1^2 + r_2^2 - 1}{2r_1 r_2}, \quad (5)$$

obtained by solving (1) for the c_j (when the r_j are all nonzero). This transformation causes S to factor as

$$S = \Omega^2 H / 256 R_1^4 R_2^4 R_3^4, \quad (6)$$

where

$$\Omega = 1 + R_1 R_2 + R_2 R_3 + R_3 R_1 - R_1^2 - R_2^2 - R_3^2, \quad (7)$$

and where H is a rather complicated eighth degree polynomial in R_1, R_2 and R_3 . Moreover, the Jacobian determinant of the transformation (5) is

$$J = \Omega / 4r_1^3 r_2^3 r_3^3. \quad (8)$$

Section 2 describes the singular situation that results when two solutions coalesce to form a double solution, causing J to vanish. This is "singular" in the sense that transformation (5) from (r_1, r_2, r_3) to (c_1, c_2, c_3) becomes locally non-invertible.

The principal result of this article is next presented, an efficient algorithm called as "DSA" for handling double solutions. Section 2 also discusses the results of experiments conducted using this algorithm. Section 3 studies the transformation (5), from the r_j to the c_j , in more detail, and lays the mathematical foundation for DSA.

¹A Mathematica notebook is available upon request.

2 DOUBLE SOLUTIONS

2.1 Double Solutions as Error Sources

This article is concerned with the situations where $\Omega = 0$ (hence $J = S = \Delta = 0$), and where $|\Omega|$ is sufficiently close to zero to cause trouble. Since J tends to be small when $|\Omega|$ is small, computational errors can result in large errors when computing the values of the r_j from those of the c_j . This situation occurs when two solutions to the quadratic system (1) coalesce or nearly coalesce into a double solution. The case where $\Omega = 0$ was introduced and studied in (Smith, 1965) and (Thompson, 1966), and later considered by others, such as (Zhang and Hu, 2005) and (Zhang and Hu, 2006). It turns out that $\Omega = 0$ corresponds to having a physical setup in which the camera's focal point is on a special circular cylinder, customarily known as the "danger cylinder."

When $S = 0$, it can be shown that $\Omega = 0$ for some solution to (1), which is thus a repeated solution. By determining that $|S|$ is smaller than some given tolerance, and then behaving as though $S = 0$, the (nearly) repeated solution can be computed more efficiently and reliably than would otherwise be the case. Rather than solving Grunert's complicated quartic polynomial, or following any of several known equivalent approaches, one only needs to follow a simple algorithm, detailed in the next subsection. As will be seen, this only requires a few basic computations, involving nothing more complicated than square roots.

There are a couple reasons why behaving as though $S = 0$, when $|S|$ is small, might be prudent. Imprecisions in measuring the c_j and/or roundoff error in computing S , mean that it might be impossible to know for certain if S is zero, positive, negative, or even non-real. Since the discriminant of the quartic polynomials involves S as a factor, it is possible that two nearly equal real solutions (or a double solution) are erroneously perceived to be complex solutions instead, and thereby ignored as being physically unrealistic. Even when two nearly equal real solutions are discovered, these are likely to be rather far from the correct solutions, owing to the small value of the Jacobian determinant.

2.2 Double Solutions Algorithm (DSA)

The following algorithm has been found to be a simple way to mitigate the difficulties caused by double solutions:

1. Receive (c_1, c_2, c_3) as input.
2. If necessary, negate any two of c_1, c_2 and c_3 , so as to make $c_1 + c_2 + c_3 \geq \frac{1}{2}$. If this is not possi-

ble, then quit, indicating that there is no repeated solution.

3. Compute σ, τ, χ, T and S , using formulas (4).
4. If $|S|$ is sufficiently small, then behave as though $S = 0$, and continue this algorithm; otherwise quit, indicating that there is no repeated solution.
5. Solve for u and v , using formulas (13). These formulas uniquely determine a u with $u \geq 0$, and a v with $|v| \leq 1$ (as can be proved).
6. Compute tentative values for r_1, r_2 and r_3 using formulas (10), and $r_j = \sqrt{R_j}$ ($j = 1, 2, 3$).
7. Compute corresponding values for c_2 and c_3 using formulas (5). Call these c'_2 and c'_3 though.
8. Test to see whether or not swapping c'_2 and c'_3 would cause them to be closer to the values of c_2 and c_3 (from step 2). If so, then swap r_2 and r_3 .
9. If any negation took place in step 2, then compensate for this by now negating a corresponding r_1, r_2 or r_3 . Negate r_1 if c_2 and c_3 were negated; negate r_2 if c_1 and c_3 were negated; negate r_3 if c_1 and c_2 were negated.
10. Return the repeated solution (r_1, r_2, r_3) .

Note that system (1) (using altered or unaltered c_j) has a repeated solution if and only if $S = 0$, and except in some very special cases, a repeated solution is only a double solution. Also, “closeness” in step 8 might be decided by considering $(c'_2 - c_2)^2 + (c'_3 - c_3)^2$ versus $(c'_2 - c_3)^2 + (c'_3 - c_2)^2$. Although the correctness of this algorithm is not proven here, the mathematical analysis that led to it is described in Section 3. Additionally, the simulations to be discussed next attest to its correctness as well.

2.3 Simulations

Simulations confirm the advantages of using the Double Solution Algorithm when $|S|$ is small. These simulations were performed using compiled Mathematica functions, running on an Intel Core Duo processor. Thus the floating point computations were performed using 64-bit IEEE floating point format. Even more dramatic results can be expected in a 32-bit floating point environment.

A radius-one danger cylinder was used. Five different distance ranges along the cylinder axes were explored: 0-2, 2-4, 4-6, 6-8 and 8-10. A camera focal point on the cylinder (within the given range) was randomly selected, and the cosines c_1, c_2, c_3 computed. DSA was tested against Grunert’s quartic polynomial method, and the resulting computed distances for r_1 were compared with the actual value of r_1 .

Next, each of the three cosines was randomly perturbed by adding or subtracting up to one one-millionth to/from it, and the two methods were compared again using the resulting data. This was again repeated, but using a maximum adjustment of one one-hundredth, rather than one one-millionth, for each cosine. In this way, fifteen different experiments (five distance ranges times three maximum cosine perturbation amounts) were considered. Each of these experiments was performed one hundred thousand times, and the results of these trials were averaged.

When the computed cosines (c_1, c_2, c_3) for a point (essentially) on the danger cylinder were left unperturbed, the ratio of the average errors using Grunert’s method versus DSA was between a hundred million and a billion. Admittedly though, the likelihood of having the camera’s focal point right on the danger cylinder, within the computational tolerance of 64-bit floating point arithmetic, is very small. Thus further experiments were conducted using slightly altered value of the cosines.

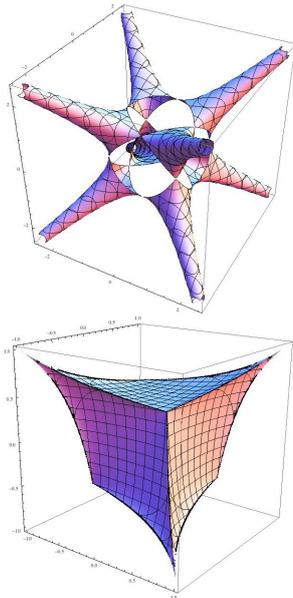
When the cosines were randomly perturbed by an amount up to one one-millionth, the ratio of the average computed errors was as much as 52, when the focal point was close to the reference point (the 0-2 range). But this ratio dropped to 14 when the focal point was far away (8-10 range).

When the cosines were randomly perturbed by an amount up to one one-hundredth, the error ratio ranged between one and two. Thus the improvement using DSA was modest in this case. Once again though, computations performed using 32-bit arithmetic, instead of 64-bit arithmetic, would more dramatically demonstrate a difference in accuracy between the two methods.

The ratio of the execution times for the two methods were also compared. Here though, it was difficult to know how much of the timing reported by Mathematica was attributable to the overhead involved in calling compiled functions from within the Mathematica interpreter. In every case, the reported speedup (ratio) was in excess of four. However, a quick check of the actual computations involved in the two methods suggests that the true speedup should be considerably higher.

3 MATHEMATICAL ANALYSIS

This section captures much of the reasoning underlying DSA. The phrases “ R -space,” “ r -space” and “ c -space” will be used to refer to the abstract three-dimensional spaces of (R_1, R_2, R_3) points, (r_1, r_2, r_3)


 Figure 1: The critical surfaces \hat{Q} and \hat{S} .

points, and (c_1, c_2, c_3) points, respectively. Because the coordinates of the points in c -space represent the cosines of angles in physical space, we are particularly interested in the points (c_1, c_2, c_3) whose coordinates have absolute value less than or equal to one.

Let \hat{T} denote the set of points in c -space for which $|c_j| \leq 1$ ($j = 1, 2, 3$) and $T \geq 0$ (another physical requirement). The boundary of this region (where $T = 0$) is shaped like an “inflated tetrahedron,” basically resembling an over-stuffed tetrahedral pillow. \hat{T} consists of this surface plus its interior. Let \hat{S} denote the set of all points in \hat{T} satisfy the equation $S = 0$. This surface is shown on the right in Figure 1. It resembles a deformed cube with two types of vertices, all of which are also on the boundary of \hat{T} . \hat{S} can be decomposed into four identical sections, each resembling a deformed triangular cone. One of these consists of points satisfying $c_1 + c_2 + c_3 \geq \frac{1}{2}$, referred to as the “principal cone.”

In r -space, there are also restrictions on the set of points (r_1, r_2, r_3) that are realizable, given the setup in physical space. Since the reference points are a distance one apart, it is immediately clear that no two of r_1, r_2 and r_3 can differ by more than one. There is an additional restriction though, imposed by the fact that the tetrahedron, in physical space, whose vertices are the camera’s focal point and the three reference points, must have positive volume. Using the Cayley-Menger determinant, it can be seen that 144 times this volume equals $\Omega + R_1 + R_2 + R_3 - 2$. So it is required that this be non-negative. Also, a quick check establishes that when the substitution (5) is used to express

T in terms of the r_j , one obtains

$$T = \frac{\Omega + R_1 + R_2 + R_3 - 2}{4R_1R_2R_3}. \quad (9)$$

Let $\hat{\mathcal{R}}$ denote the region in r -space consisting of points (r_1, r_2, r_3) satisfying $|r_1 - r_2| \leq 1$, $|r_2 - r_3| \leq 1$, $|r_3 - r_1| \leq 1$ and $\Omega + R_1 + R_2 + R_3 \geq 2$. These are the points in r -space that are physically realizable, assuming that negative values of r_j are admissible.

Let \hat{Q} denote the subset of $\hat{\mathcal{R}}$ consisting of points for which $\Omega = 0$. This surface is shown on the left in Figure 1. Because of (6), it is clear that the points of \hat{Q} (in r -space) correspond to points of \hat{S} (in c -space), under the transformation (5). It can be shown that this mapping from \hat{Q} to \hat{S} is onto, except for a set of measure zero.

From equation (7), observe that the surface \hat{Q} corresponds to a portion of a circular cylinder in R -space. Only a portion of the cylinder is admissible though, due to the restriction that $R_1 + R_2 + R_3 \geq 2$, by (9), since $T \geq 0$ and $\Omega = 0$. The semi-cylinder (surface) can be parameterized as follows:

$$\begin{cases} R_1 &= \frac{1}{3}(2 + u - 2\cos\theta), \\ R_2 &= \frac{1}{3}(2 + u + \cos\theta + \sqrt{3}\sin\theta), \\ R_3 &= \frac{1}{3}(2 + u + \cos\theta - \sqrt{3}\sin\theta). \end{cases} \quad (10)$$

Here $u = R_1 + R_2 + R_3 - 2$, which must be non-negative, and we will assume too that $0 \leq \theta < 2\pi$. Notice that replacing θ with $\theta \pm 2\pi/3 \pmod{2\pi}$ induces a cyclic permutation of $\{R_1, R_2, R_3\}$. Replacing θ with $2\pi - \theta$ swaps R_2 and R_3 . It is helpful to use the notation $v = \cos\theta$ and $w = \pm\sqrt{1 - v^2} = \sin\theta$.

Because of the independent sign choices for each of r_1, r_2 and r_3 , for given R_1, R_2 and R_3 , there are eight identical sections, or “legs,” of \hat{Q} in r -space. These correspond to the same semi-cylinder in R -space, with $u \geq 0$. The section for which r_1, r_2 and r_3 are all non-negative will be called the “principal leg” of \hat{Q} . The points on the principal leg of \hat{Q} get mapped, via (5), to points on the principal cone of \hat{S} . Expressing points on the principal leg in terms of u, v and w , the mapping (5) becomes the following:

$$\begin{aligned} c_1 &= \frac{1 + 2u + 2v}{2\sqrt{2 + u + v + \sqrt{3}w}\sqrt{2 + u + v - \sqrt{3}w}}, \\ c_2 &= \frac{1 + 2u - v - \sqrt{3}w}{2\sqrt{2 + u - 2v}\sqrt{2 + u + v - \sqrt{3}w}}, \\ c_3 &= \frac{1 + 2u - v + \sqrt{3}w}{2\sqrt{2 + u - 2v}\sqrt{2 + u + v + \sqrt{3}w}}. \end{aligned} \quad (11)$$

These formulas then lead immediately to the following formulas:

$$\begin{aligned}\sigma &= \frac{6(1+3v-4v^3)+3u(3+12u+4u^2)}{4(2+u-2v)(1+4u+u^2+4v+4v^2+2uv)}, \\ \tau &= \frac{(1+2u+2v)(-1+2u+2u^2-v+2v^2-2uv)}{4(2+u-2v)(1+4u+u^2+4v+4v^2+2uv)}, \\ \chi &= \frac{3(1+3v-4v^3)^2+3u(1+3v-4v^3)(3+3u+4u^2)+6u^3(3+u)(3+6u+2u^2)}{4(1+4u+u^2+4v+4v^2+2uv)^2}.\end{aligned}\quad (12)$$

As complicated as formulas (11) and (12) are, surprisingly simple inversion formulas exist. Given, c_1, c_2 and c_3 , and using (4) to compute σ and τ , the values of u and v can be readily deduced from the following:

$$\begin{aligned}(1+u)^2 &= \frac{3(1+8\tau)}{4(1+2\tau-\sigma)}, \\ (1+v)^2 &= \frac{3(1+2\tau-3c_2^2)(1+2\tau-3c_3^2)}{4(1+2\tau-3c_1^2)(1+2\tau-\sigma)}.\end{aligned}\quad (13)$$

These two formulas can be efficiently confirmed using mathematical manipulation software, by making substitutions using (11) and (12).

4 CONCLUSIONS

The *Double Solution Algorithm (DSA)* presented in this article gives a very practical, very fast, and highly accurate method for solving the P3P problem, when dealing with the setup where the camera's focal point is on or near the danger cylinder, and when the reference points are equidistant from each other. It relies on the inversion of a transformation between two surfaces. Surprisingly, the inverse mapping turns out to be simpler to compute than the original mapping. It would be quite useful to find a generalization of DSA to the situation where the reference points are no longer assumed to be equidistant from each other.

REFERENCES

Chen C-S, Hung Y-P, Shih S-W, Hsieh C-C, Tang C-Y, Yu C-G and Chang Y-C (1998). Integrating virtual objects into real images for augmented reality. In *VRST'98, ACM Symp. Virtual Reality Software and Technology*, pp. 1-8. ACM.

- Faugère J-C, Moroz G, Roullier F, El Din M S (2008). Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities. In *ISSAC'08, 21st ACM Int. Symp. Symbolic and Algebraic Computation*, pp. 79-86. ACM.
- Gao X-S, Hou X-R, Tang J, and Cheng H-F (2003). Complete solution classification for the perspective-three-point problem. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 25, n. 8, pp. 930-943. IEEE.
- Grunert J A (1841). Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. In *Grunerts Archiv für Mathematik und Physik*, Band 1, pp. 238-248. Verlag von C. A. Koch.
- Haralick R M, Lee C-N, Ottenberg K and Nölle N (1994). Review and analysis of solutions of the three point perspective pose estimation Problem. In *J. Computer Vision*, v. 13, n. 3, pp. 331-356. Springer Netherlands.
- Merritt, E L (1949). Explicit three-point resection in space. In *Photogrammetric Engineering*, v. 15, n. 4, pp. 649-655. Amer. Soc. Photogrammetry.
- Müller F J (1925). Direkte (exakte) lösung des einfachen rückwärtsein-schneidens im raume. In *Allegemaine Vermessungs-Nachrichten*. Wichmann Verlag.
- Ohayon S and Rivlin E (2006). Robust 3D head tracking using camera pose estimation. In *ICIP'06, Int. Conf. Image Processing*, pp. 1063-1066. IEEE.
- Qingxuan J, Ping Z and Hanxu S (2006). The study of positioning with high-precision by single camera based on P3P algorithm. In *INDIN'06, IEEE Int. Conf. Industrial Informatics*, pp. 1385-1388. IEEE.
- Smith A D N (1965). The explicit solution of the single picture resolution problem, with a least squares adjustment to redundant control. In *Photogrammetric Record*, v. 5, n. 26, pp. 113-122. Wiley-Blackwell.
- Thompson E H (1966). Space resection: failure cases. In *Photogrammetric Record*, v. 5, n. 27, pp. 201-204. Wiley-Blackwell.
- Wolfe W J, Mathis D, Sklair C W, and Magee M (1991). The perspective view of three points. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 13, n. 1, pp. 66-73. IEEE.
- Zhang C-X and Hu Z-Y (2005). A general sufficient condition of four positive solutions of the P3P problem. In *J. Comput. Sci. & Technol.*, v. 20, n. 6, pp. 836-842. Springer.
- Zhang C-X and Hu Z-Y (2006). Why is the danger cylinder dangerous in the P3P problem? In *Acta Automatica Sinica*, v. 32, n. 4, pp. 504-511. Elsevier.