ALGORITHMS FOR SOLVING THE FOUR POINT MOTION PROBLEM IN COMPUTER VISION.

JOHAN PHILIP

ABSTRACT. We present two algorithms for solving the motion problem for four coplanar points using a calibrated camera. The accuracy of the algorithm for noisy data is compared with the accuracy of algorithms for more points.

1. INTRODUCTION

The motion problem in computer vision consists in determining the motion of a rigid object. Two perspective images are taken of the object, one before and one after the motion. The input data are a number of corresponding points on the object that have been recognized in the two images.

By a solution we mean the successful calculation of all motions that comply with the perspective coordinates. If there is more than one solution, we must use 'other information' for selecting the calculated motion corresponding to the true motion.

Our main topic is the case that four points have been recognized and that they are known to be coplanar on the object. This information is sufficient for solving the motion problem and there are in general two motions that comply with the data.

Our equations are deduced from the fact that the distance between two points of the object is the same before and after the motion.

The accuracy of the solution when the data are noisy is important. We report results from Monte Carlo tests for comparing the accuracy obtainable with four points on the object with that obtainable with more points.

The motion problem has a long history in photogrammetry, where it is called the relative orientation problem. Our attack on the four point problem resembles that of Hofmann-Wellenhof [4]. See also Wunderlich [16] and Triggs [13]. The problem is also studied in Faugeras [1], Hartley [3], Longuet-Higgins [5], Maybank [6], and Negahdaripour [7].

In the accuracy comparison, we use algorithms for five and more points. These algorithms don't require coplanarity of the points. The five-point problem has up to ten solutions. More than six solutions

Date: December 12, 2003.

is extremely rare [1]. Using this algorithm, one has to resort to some method of choosing the 'correct' solution. Algorithms for the five-point problem are presented in Philip [10], Nistér [8], and Triggs [14]. For six or more points, the solution is unique. Algorithms for the six-point problem are in Hofmann-Wellenhof [4] and Philip [10]. The seven-point algorithm is in Philip [11]. The eight-point algorithm which handles eight or more points is given by Stefanovic [12]. The numerical treatment of this algorithm was improved by Tsai and Huang [15]. The eight-point algorithm has over the years been the topic of numerous papers.

2. Notation and formulation.

We consider a pinhole camera and denote the 3D coordinates of the points before and after the motion by $\mathbf{U}^i$ and $\mathbf{V}^i$, $(1 \leq i \leq 4)$, respectively. These coordinates are not observed but only their perspective coordinates

$$\mathbf{u}^i = \begin{pmatrix} {U^i}_1/{U^i}_3 \\ {U^i}_2/{U^i}_3 \\ 1 \end{pmatrix} \qquad \mathbf{v}^i = \begin{pmatrix} {V^i}_1/{V^i}_3 \\ {V^i}_2/{V^i}_3 \\ 1 \end{pmatrix}.$$

Any rigid motion can be described by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$

$$\mathbf{V^i} = \mathbf{R}\mathbf{U}^i + \mathbf{t}.$$

Our present problem is to determine $\mathbf{R}$ and $\mathbf{t}$ given that the four points U^i are known to be sitting in a plane. Since the motion is rigid, also the four V^i are coplanar. Since we only know the perspective coordinates, neither the size of the object nor the length of $\mathbf{t}$ can be recovered.

3. Deduction of equations.

Start by determining the barycentric coordinates α_i and β_i so that

$$\sum_{i=1}^{4} \alpha_i \mathbf{u}^i = \mathbf{0} \qquad \sum_{i=1}^{4} \beta_i \mathbf{v}^i = \mathbf{0}. \tag{1}$$

If $\alpha_i = 0$, for one index, say k, the $\mathbf{u}^i$ corresponding to the other indices are collinear, and the problem cannot be solved. If this happens, also $\beta_k = 0$.

Since the third component of $\mathbf{u}^i$ and $\mathbf{v}^i$ are one, we have also

$$\sum_{i=1}^{4} \alpha_i = 0 \qquad \sum_{i=1}^{4} \beta_i = 0.$$

[1] David Nister, <dnister@sarnoff.com>, has provided me with the only example I have seen of a five-point problem with ten solutions.

These two relations are of no importance. We could alternatively have normalized the $\mathbf{u}^i$ and $\mathbf{v}^i$ to have euclidean length one, in which case these relations would not be true.

Having homogeneous equations, we need a normalization of the solutions. We achieve this by putting $\alpha_4 = \beta_4 = -1$ so the calculations amount to solving two linear 3×3 systems. If these systems turn out to be singular, we have the situation that the problem is insolvable because $\alpha_4 = \beta_4 = 0$.

The subsequent calculations aim at determining the true positions of the points in 3D, that is to determine the ${U_3}^i$ and ${V_3}^i$ (up to a common scale factor).

We will use the notations $\lambda_i = {U_3}^i$ and $\mu_i = {V_3}^i$ so that the 3D points are $\lambda_i \mathbf{u}^i$ and $\mu_i \mathbf{v}^i$. These points are assumed to be in front of the camera so that $\lambda_i > 0$ and $\mu_i > 0$.

Equation (1) leads to

$$(2) \qquad \sum_{i=1}^{4} (\alpha_i/\lambda_i)\lambda_i \mathbf{u}^i = \mathbf{0} \qquad \sum_{i=1}^{4} (\beta_i/\mu_i)\mu_i \mathbf{v}^i = \mathbf{0}.$$

The α_i/λ_i and the β_i/μ_i are the barycentric coordinates for points known to sit in a plane, so we have

$$(3) \qquad \sum_{i=1}^{4} \alpha_i/\lambda_i = 0 \qquad \sum_{i=1}^{4} \beta_i/\mu_i = 0.$$

Moreover, these two sets are the barycentric coordinates of the same point configuration so there is a nonzero proportionally constant k such that

$$(4) \qquad \alpha_i/\lambda_i = k\beta_i/\mu_i \quad \text{for} \quad 1 \leq i \leq 4.$$

Introducing the notation $\kappa_i = \beta_i/\alpha_i$, we can write (4) in the form

$$(5) \qquad \mu_i = k\kappa_i\lambda_i.$$

The distance between point i and point j on the object is the same before and after the motion so we have

$$(6) \qquad \|\lambda_i \mathbf{u}^i - \lambda_j \mathbf{u}^j\|^2 = \|\mu_i \mathbf{v}^i - \mu_j \mathbf{v}^j\|^2.$$

Inserting the μ_i of (5) in (6) and expanding the norms as scalar products gives

$$(7) \qquad \lambda_i^2[\|\mathbf{u}^i\|^2 - k^2{\kappa_i}^2\|\mathbf{v}^i\|^2] \\ - 2\lambda_i\lambda_j[(\mathbf{u}^i, \mathbf{u}^j) - k^2\kappa_i\kappa_j(\mathbf{v}^i, \mathbf{v}^j)] \\ + \lambda_j^2[\|\mathbf{u}^j\|^2 - k^2{\kappa_j}^2\|\mathbf{v}^j\|^2] = 0.$$

Introduce the following linear functions of k^2

$$(8) \qquad c_{ij} = (\mathbf{u}^i, \mathbf{u}^j) - k^2\kappa_i\kappa_j(\mathbf{v}^i, \mathbf{v}^j).$$

Notice that the κ_i have been calculated and that the inner products can be calculated from the input data.

Writing up equations (7) for $1 \leq i < j \leq 3$ with these notations, we get

$$c_{11}{\lambda_1}^2 - 2c_{12}\lambda_1\lambda_2 + c_{22}{\lambda_2}^2 = 0 \tag{9}$$

$$c_{11}{\lambda_1}^2 - 2c_{13}\lambda_1\lambda_3 + c_{33}{\lambda_3}^2 = 0 \tag{10}$$

$$c_{22}{\lambda_2}^2 - 2c_{23}\lambda_2\lambda_3 + c_{33}{\lambda_3}^2 = 0. \tag{11}$$

The subsequent calculations will be based on these three equations. Since the four points are linearly dependent, we get no further linearly independent equation by using point four in (7).

4. First method of solution.

Use the c_{ij} for $1 \leq i, j \leq 3$ to form the symmetric matrix C. Notice that the off-diagonal elements of C have opposite signs to the corresponding terms in (9) - (11). We show in the appendix that the system (9) - (11) has a solution with nonzero λ_i only if

$$\det C = 0. \tag{12}$$

Since each entry of the determinant is a linear function of k^2, (12) is a cubic equation in k^2, which must hold if there shall exist a solution with nonzero λ_i. Step one of the solution process is to solve this cubic. In the course of describing our second method, we will show that the solution has three positive real roots and that it is the middle one that corresponds to the solution of our geometric problem. Inserting this middle k^2 in the c_{ij} of equations (9) and (10), we get two quadratic equations for λ_2/λ_1 and λ_3/λ_1, respectively. The solutions are normalized by putting $\lambda_1 = 1$. Of the four solution pairs (λ_2, λ_3) only two can satisfy the second order equation (11). These two pairs will produce two tentative solutions to our motion problem. Knowing λ_1 - λ_3, we can use (3) to calculate λ_4. If all four λ_i are positive, we have a solution in front of the camera.

5. Second method of solution.

Form a 3×3 - matrix U with columns $\mathbf{u}^i$, $1 \leq i \leq 3$. Form V in the same way with $\kappa_i \mathbf{v}^i$. Since the the $\mathbf{u}^i$ are assumed to be linearly independent and the same holds for the $\mathbf{v}^i$, these matrices are nonsingular. With these notations, $C = U^T U - k^2 V^T V$. The equation $detC = det(U^T U - k^2 V^T V) = 0$ in (12) is the equation for finding the generalized eigenvalues $(= k^2)$ of $U^T U$ with respect to $V^T V$, see Golub and van Loan, [2], chap. 8.6. Here, the matrices $U^T U$ and $V^T V$ are symmetric positive definite and this reference gives an explicite method for handling this case. Theorem 8.6.4 asserts the existence of an invertible matrix X such that

$$X^T C X = X^T U^T U X - k^2 X^T V^T V X = D_U - k^2 D_V, \tag{13}$$

whereD_U and D_V are diagonal. Since $U^T U$ and $V^T V$ are positve definite, the diagonal elements are positive. The construction goes as follows: form a 6×3 - matrix by stacking U on top of V and calculate the QR-decomposition of this matrix with Q having orthonormal columns. Let Q_{11} be the top 3×3 - submatrix of Q . Calculate the SVD of $Q_{11} = GSH^T$ for obtaining the singular values $s_1 \geq s_2 \geq s_3 > 0$. The six diagonal elements are $(D_U)_{ii} = {s_i}^2 > 0$ and $(D_V)_{ii} = 1 - {s_i}^2 > 0$.

The three generalized eigenvalues, which also are the solutions of (12) are

$$(D_U)_{ii}/(D_V)_{ii} = {s_i}^2/(1 - {s_i}^2) > 0 \text{ for } 1 \leq i \leq 3. \tag{14}$$

We shall show that it is the middle eigenvalue $k^2 = {s_2}^2/(1 - {s_2}^2)$ that solves our problem because the others lead to complex valued λ_i.

Define $\lambda^T = (\lambda_1, \lambda_2, \lambda_3)$ and consider the quadratic form $\phi(\lambda) = \lambda^T C \lambda$. Equations (9) - (11) are special instances of the equation $\phi = 0$. E.g. (9) has the form $\phi(\lambda_1, -\lambda_2, 0) = 0$. Using the coordinate transformation $\lambda = X\eta$ and inserting $k^2 = {s_2}^2/(1 - {s_2}^2)$, we can write

$$\begin{aligned} \phi = \lambda^T C \lambda = \eta^T X^T C X \eta = \eta^T D_U \eta - k^2 \eta^T D_V \eta = \\ ({s_1}^2 - k^2(1 - {s_1}^2)){\eta_1}^2 + ({s_3}^2 - k^2(1 - {s_3}^2)){\eta_3}^2. \end{aligned} \tag{15}$$

Since the s_i are decreasing, the coefficient for ${\eta_1}^2$ in (15) is positive and that for ${\eta_3}^2$ is negative. We get $\phi = 0$ in (15) for some η_2 and $(\eta_3/\eta_1)^2 = -\dfrac{{s_1}^2 - k^2(1 - {s_1}^2)}{{s_3}^2 - k^2(1 - {s_3}^2)} = \dfrac{{s_1}^2 - {s_2}^2}{{s_2}^2 - {s_3}^2} > 0$, which gives two real values for η_3/η_1.

Before calculating λ, we note that inserting the first eigenvalue for k^2 in (15) and requiring $\phi = 0$ would give $(\eta_3/\eta_2)^2 < 0$ and a complex η_3/η_2. The same would occur with the third eigenvalue.

To get a solution of e.g. (9), we equate the obtained ratio η_3/η_1 with that obtained from $\lambda = X\eta$ using $\lambda_3 = 0$. More precisely, theorem 8.6.4 gives the explicite expression $X = (H^T R)^{-1}$, so $\eta = H^T R \lambda$. Since $\lambda_3 = 0$, we get η_1 and η_3 as linear functions of λ_1 and λ_2. Forming the ratio η_3/η_1 and equating it with the same ratio obtained above, we get an expression for λ_1/λ_2. Switching the roles of λ_2 and λ_3, we obtain a ratio λ_1/λ_3 satisfying (10). We normalize the solution by putting $\lambda_1 = 1$ and calculate λ_4 from (3).

Notice that the $(\lambda_1, \lambda_2, \lambda_3)$ solving (9) - (11) is not an eigenvector corresponding to the chosen eigenvalue k^2.

Notice also that for instance equation (9) has the form and meaning $\phi(\lambda_1, -\lambda_2, 0) = \|\mathbf{U}^1 - \mathbf{U}^2\|^2 - \|\mathbf{V}^1 - \mathbf{V}^2\|^2 = 0$.

6. Accuracy comparison.

We have run Monte Carlo test in order to assess the accuracy achievable from four coplanar points. For comparison, these tests have been repeated for objects with more points. All objects and motions have been generated by random numbers as follows:

The focal point of the camera is in the origin of a coordinate system. The camera plane is parallell to the xy-plane and the z-axis points towards the object.

For objects having more than five points, the points are evenly spread in a cube with side = 2 having its center in a cube with side = .5 and centered in $x = y = 0$, and $z = 4$. In the four point case, the points are first generated as above and then moved along the projection rays to a plane. The normal direction of this plane is chosen in a cone which is rotation symmetric around the z-axis and has a solid angle of π. The density of normals is greater in the middle of the cone and decreases towards its boundary. We think that the limitation of the plane normals to this cone describes a more realistic situation than having normals spread over all directions. The distance to the plane is evenly distributed in (3,5).

The motion parameters are generated as follows:
The rotation angle θ is evenly spread over $0 \leq \theta \leq 30$ degrees.
The rotation axis $\mathbf{n}$ of the motion is evenly spread over all directions.
The translation $\mathbf{t}$ is evenly spread over all directions with $z \geq 0$. This condition is imposed to avoid having to reject too many examples because the moved object is behind the camera. The length of $\mathbf{t}$ is evenly spread in (0,3).

Gaussian errors with mean zero and standard deviation σ are added to the projective coordinates. We present results from tests with $\sigma = .0001$. The average size of the projective coordinates is about .2 so the relative noise is about .5 %.

The errors of the computed motion parameters are measured in degrees. They are:

(i) The error of the rotation angle $\Delta\theta$

(ii) The angle between the true and calculated rotation axis $\Delta\mathbf{n}$

(iii) The angle between the true and calculated translation direction $\Delta\mathbf{t}$.

The results are presented in Figure 1. We have chosen to present medians of errors instead of averages. The reason is the better stability of the medians. The averages are strongly affected by a few very large errors which always occur in these tests.

Since the four-point problem generally has 2 solutions and the five-point problem has up to six solutions, the 'correct' solution must be chosen in some way. This causes no problem in the Monte Carlo tests,

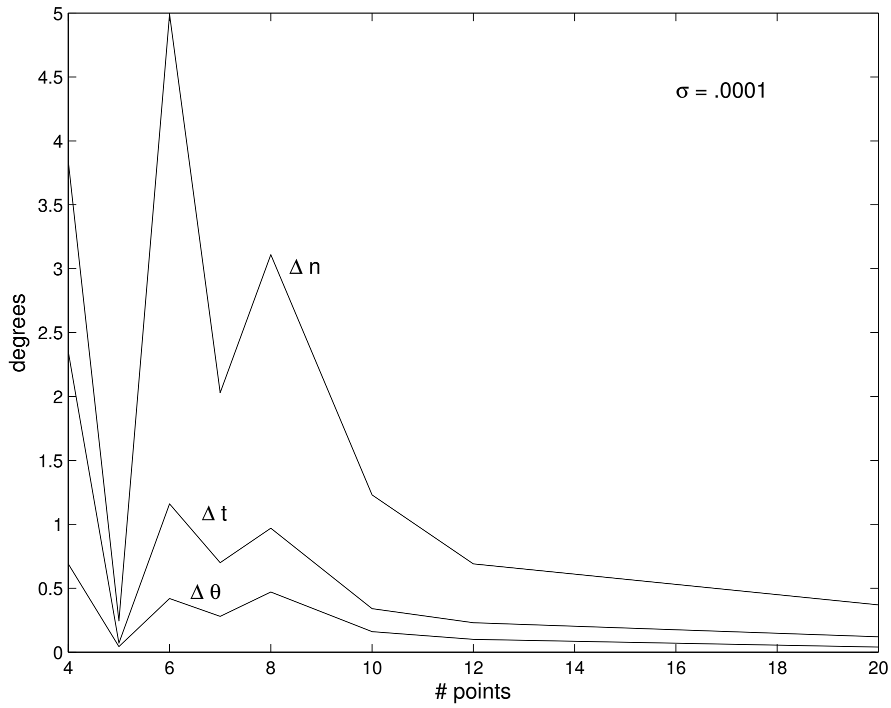

FIGURE 1. Medians of errors in degrees of (i) rotation angle θ, (ii) rotation axis $\mathbf{n}$, (iii) translation direction $\mathbf{t}$. The added gaussian noise has $\sigma = .0001$. The average size of the projective coordinates is about .2 so the relative noise is about .5%.

where we know the true motion. See e.g. Nister [9] about how this can be done in practice.

The most striking result is that the five-point algorithm is so resilient to noise. The seven-point algorithm is better than the eight-point algorithm.

The obtainable accuracy is depending on the size of the solid angle subtended by the convex hull of the object. With our method of generating the objects in a cube of fixed size, the size of the objects increases with the number of points. In order to get a more fair comparison of the methods, we have also used varying size of the cube so that all objects, in the average, subtend the same solid angle. The results presented in Figure 2 are for objects having an average solid angle equal to .1 .

7. DISCUSSION

In the Monte Carlo tests with four coplanar points, we have tried the two algorithms of this paper and the one given by Triggs [13]. Without added noise, they all give the 'true' solution and sometimes

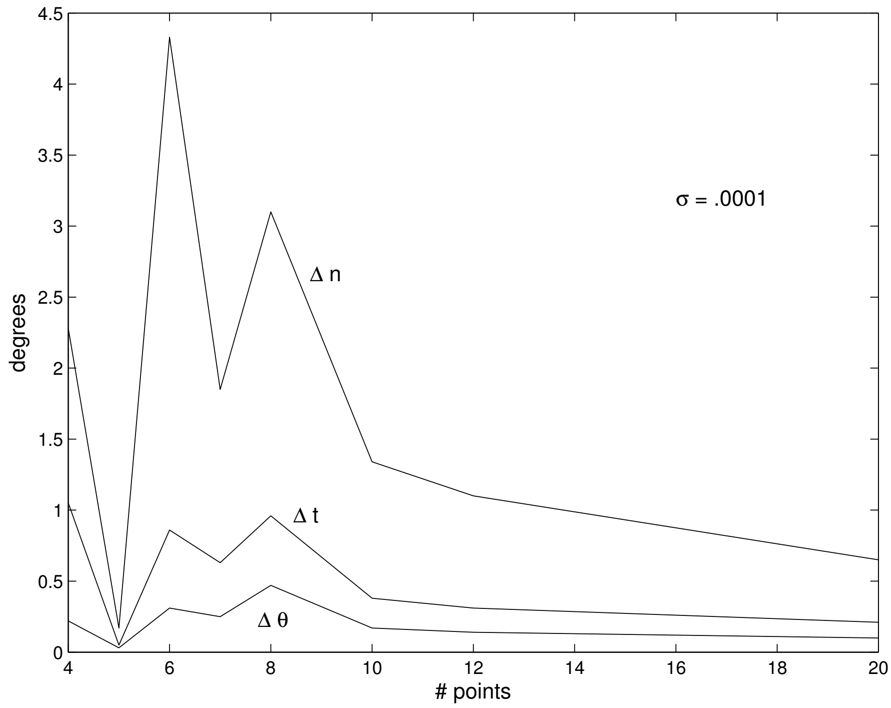

FIGURE 2. Medians of errors in degrees of (i) rotation angle θ, (ii) rotation axis $\mathbf{n}$, (iii) translation direction $\mathbf{t}$. The added gaussian noise has $\sigma = .0001$. For each #points, the average size of the objects is adjusted so that the solid angle of its convex hull equals .1. The average size of the projective coordinates is about .2 so the relative noise is about .5%.

another. Triggs method always gives a second solution, which sometimes is behind the camera. The noise amplification is about the same for all three methods. We show the relative computation times for the various methods in Table 1

The methods for six and more points produce besides the solution one or more test variables, which indicate the accuracy of the solution, see e.g. [11]. In the four point algorithms, the values of the α_i and β_i give some indication about how far the problem is from being critical. If some of the computed α_i or β_i is very small or all α_i and β_i are very large, three of the points are almost collinear. We tested skipping generated test problems having some α_i or $\beta_i \leq .1$ or ≥ 10. This reduced both the average and median errors by about one half. About one third of the tests were skipped. We also tried skipping problems for which the convex hull of the test object had an area in the projective plane less than .03. This is essentially the same as solid angle less than .03. This reduced the average errors because a few tests with really big

#points	Algorithm	time
4	algorithm I	8
4	algorithm II	10
4	Triggs [13]	11
5		145
6		47
7		25
8		24
10		28
12		29
20		46

TABLE 1. Relative computation time

errors were eliminated. We suggest that both these conditions on the object are used in practice.

In setting up equations (9) - (11), three of the four points are used. Here, one could think of selecting those three of the four points that give the most stable calculations. We have tried this and found that that nothing was gained by selecting points. The reason is that the numerical accuracy is much greater than the accuracy of the noisy input data.

Our method of generating the plane that the four points sit in favours planes up front. We tried also plane normals evenly spread over all directions. This decreased the average errors of the computed parameters. This means that more tilted planes give better estimates.

8. Appendix

Suppose there exists λ_1,λ_2, and λ_3, all different from zero satisfying (9) - (11). We shall show that this implies $det(C) = 0$. The following calculations can be carried out if the $\lambda_i \neq 0$.

Form the difference between (9) and (10) to eliminate their first terms:

$$2\lambda_1(c_{13}\lambda_3 - c_{12}\lambda_2) = c_{33}{\lambda_3}^2 - c_{22}{\lambda_2}^2. \tag{16}$$

Form the difference $c_{33}{\lambda_3}^2$ (9)$-c_{22}{\lambda_2}^2$(10) to eliminate their last terms:

$${\lambda_1}^2 c_{11}(c_{33}{\lambda_3}^2 - c_{22}{\lambda_2}^2) = 2\lambda_1\lambda_2\lambda_3(c_{12}c_{33}\lambda_3 - c_{13}c_{22}\lambda_2). \tag{17}$$

Solve both (16) and (17) for λ_1

$$\lambda_1 = \frac{c_{33}{\lambda_3}^2 - c_{22}{\lambda_2}^2}{2(c_{13}\lambda_3 - c_{12}\lambda_2)} = \frac{2\lambda_2\lambda_3(c_{12}c_{33}\lambda_3 - c_{13}c_{22}\lambda_2)}{c_{11}(c_{33}{\lambda_3}^2 - c_{22}{\lambda_2}^2)}. \tag{18}$$

Multiply crosswise by the denominators:

$$(19)\quad c_{11}{c_{22}}^2{\lambda_2}^4 - 4c_{12}c_{13}c_{22}{\lambda_2}^3\lambda_3 + \\ + (4{c_{12}}^2c_{33} + 4{c_{13}}^2c_{22} - 2c_{11}c_{22}c_{33}){\lambda_2}^2{\lambda_3}^2 - \\ - 4c_{12}c_{13}c_{33}\lambda_2{\lambda_3}^3 + c_{11}{c_{33}}^2{\lambda_3}^4 = 0.$$

Subtract c_{11} times the square of (11) from (19) to get rid of the ${\lambda_2}^4$ and ${\lambda_3}^4$ terms :

$$(20)\quad (-4c_{12}c_{13} + 4c_{11}c_{23})c_{22}{\lambda_2}^3\lambda_3 + \\ + (4{c_{12}}^2c_{33} + 4{c_{13}}^2c_{22} - 4c_{11}c_{22}c_{33} - 4c_{11}{c_{23}}^2){\lambda_2}^2{\lambda_3}^2 + \\ + (-4c_{12}c_{13} + 4c_{11}c_{23})c_{33}\lambda_2{\lambda_3}^3 = 0.$$

Divide out $4\lambda_2\lambda_3$ from (20) and subtract it by $(-c_{12}c_{13} + c_{11}c_{23})$ times (11) :

$$({c_{12}}^2c_{33} + {c_{13}}^2c_{22} - c_{11}c_{22}c_{33} - c_{11}{c_{23}}^2 + 2c_{23}(-c_{12}c_{13} + c_{11}c_{23}))\lambda_2\lambda_3 = 0.$$

This turns out to equal minus the determinant of c times $\lambda_2\lambda_3$. Since λ_2 and λ_3 are assumed to be nonzero, the determinant must be zero.

References

[1] O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, ISBN 0-262-06158-9, 1993.

[2] G. Golub and C. van Loan, *Matrix Computations*, North Oxford Academic, Oxford 1983.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN 0-521-62304-9, 2000.

[4] B. Hofmann-Wellenhof, Die gegenseitige Orientierung von zwei Strahlenbündeln bei übereinstimmung, bei unbekannten Näherungswerten und durch ein nichtiteratives Verfahren, *Deutsche Geodätische Kommision bei der Bayerischen Akadamie der Wissenschaften*, Reihe C, Heft Nr 257, 1979.

[5] H. Longuet-Higgins, The Reconstruction of a Plane Surface from Two Perspective Projections, *Proc. R. Soc. Lond. B*, 277:399-410, 1986.

[6] S. Maybank, *Theory of Reconstruction from Image Motion*, Springer-Verlag, ISBN 3-540-55537-4, 1993.

[7] S. Negahdaripour, Closed-Form Relationship Between the Two Interpretations of a Moving Plane, *J. Optical Society of America*, 7(2):279-285, 1990.

[8] D. Nistér, An Efficient Solution to the Five-Point Relative Pose Problem, *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 195-202, 2003.

[9] D. Nistér, Preemptive RANSAC for Live Structure and Motion Estimation, *IEEE International Conference on Computer Vision*, pp. 199-206, 2003.

[10] J. Philip, A non-iterative algorithm for determining all essential matrices corresponding to five point pairs, *Photogrammetric Record* 15(88), pp. 589-599, October 1996.

[11] J. Philip, Critical point configurations of the 5-, 6-, 7-, and 8-point algorithms for relative orientation, Technical Report, TRITA-MAT-1998-MA-13 (Februari 1998), www.math.kth.se/~johanph

[12] P. Stefanovic, Relative orientation - a new approach *I.T.C. Journal*, 1973-3, pp. 417-448.

[13] B. Triggs. SVD based Relative Orientation from Planar Scenes. www.inrialpes.fr/movi/people/Triggs

[14] B. Triggs. Routines for Relative Pose of Two Calibrated Cameras from 5 Points. Technical Report. www.inrialpes.fr/people/Triggs, INRIA, France, 2000.

[15] R.Y. Tsai and T.S. Huang, Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces, *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 1984, 6(1), pp. 13-27.

[16] W. Wunderlich, Rechnerische rekonstruktion eines ebenes objekts aus zwei photographien, *Mitteilungen Geodät. Inst. Graz*, Folge 40 (Festschrift K. Rinner zum 70 Geburstag) pp. 365-77, June 1995

Department of Mathematics, Royal Institute of Technology, S-10044 Stockholm Sweden

E-mail address: `johanph@math.kth.se`