Linear N 4-point pose determination

Long Quan, Zhong-Dan Lan

Linear $N \geq 4$-Point Pose Determination

Long QUAN Zhongdan LAN

CNRS-GRAVIR-INRIA, ZIRST – 655 avenue de l'Europe, 38330 Montbonnot, France
Long.Quan@imag.fr, Zhongdan.Lan@imag.fr

Abstract

The determination of the position and the orientation of the camera from the known correspondences of the reference points and the image points is known as the problem of pose estimation *in computer vision or* space resection *in photogrammetry. It is well known that using 3 corresponding points has at most 4 solutions. Less appears to be known about the cases of 4 and 5 points. In this paper, we describe linear solutions that always give the unique solution to 4-point and 5-point pose determination for the reference points not lying on the critical configurations. The same linear method can also be extended to any $n \geq 5$ points.*

The robustness and accuracy of the method are experimented both on simulated and real images.

1 Introduction

Given a set of correspondences between the reference points in space and image points, *pose estimation* consists of determining the position and orientation of the calibrated camera with respect to the known reference points. The problem is called *space resection* in photogrammetry community. This is one of the most important and also probably the oldest task in computer vision as well as in photogrammetry. There have been many methods developed either for the minimum data case or for the redundant data case.

It is well known since long time—the first algebraic solution may be tracked back to 1841 by the photogrammetrists [9]—that there is closed form solutions for 3 points. Many variants [6, 7] of the basic 3-point algorithm, which essentially use different order of substitutions and the ways the calculations are performed as summarized in [9], have been developed. The basic 3-point method involves two steps. The first step uses the depths of the points with respect to the perspective center of the camera as unknowns whose solutions are obtained by solving a four-degree polynomial. The second step involves the absolute orientation to solve a similarity transformation from three 3D-3D point correspondences. For the second step, there exist nice linear least squares solutions based on quaternion representation of rotations [11, 4]. One representative of such methods that is popular in computer vision can be found in Fischler and Bolles [6] together with a RANSAC paradigm to detect outliers of the data. They also termed the problem as the perspective-3-point-problem (P3P) or generally PnP for any n points. An iterative method due to Church [18] from 3 points for pose determination is equally widely used by the photogrammetrists as described in the manual [18]. Haralick et al. [9] reviewed many old and new variants of the basic algebraic 3-point method and carefully examined their numerical calculation stabilities.

Using a minimum of three points leads to ambiguous solutions, additional information is unavoidable to guarantee a unique solution. Generally as soon as we have at least 4 points, we could expect a unique solution if these points do not form a special configuration in space. All critical configurations of reference points with which multiple distinct or coinciding (unstable) solutions are unavoidable are known [21, 22]. The pose estimation is ambiguous if the projection center is coplanar with the reference points (any three of the four) or it goes through the euclidean horopter curve or one of its degenerate forms (the euclidean form of the horopter is living in a circular cylinder) uniquely determined by four given reference points.

Unfortunately, most of the approaches developed for redundant data case rely either on iterative methods or on applying the closed form solutions on the minimal subsets of the redundant data. The iterative methods suffer from the problems of initialisation and convergence, while applying the algebraic methods for the subsets suffer from the poor noise filtering and the difficulty on selection of the common root from the noised data. For instance, Lowe [16] and Yuan [23] developed techniques relying on the Newton-Raphson method. Recently, D. Dementhon and L. Davis [2] presented an iterative algorithm of 25 lines of code for 4 points or more starting from scaled orthography projection. One exception may be the approach of Horaud et al. [10] in which P4P is converted into a special 3-line problem. Unfortunately, this conversion uses only partial information and inherits the same 4-degree polynomial equations.

The calibration procedure is closely related to the pose estimation, but strictly speaking different since the calibration estimates simultaneously the pose and the intrinsic parameters of the camera. Abdel-Aziz and Karara [18], Sutherland [20] and Ganapathy [8] proposed the direct linear method for solving the 11 entries of the projection matrix of the camera (DLT method) from at least 6 correspondence points. The method is further improved by Faugeras and Toscani [5] using different constraint on the projection matrix. Lenz and Tsai [12] proposed both linear and nonlinear calibration methods. Although these methods might be applied to the pose determination, generally the pose should be done with less than 6 points and the calibration viewd this way is a heavy over-parameterization for the pose. Mostly in computer vision, methods for pose estimation using line segments instead of points as image features were also developed. Dhome et al. [3] and Chen [1] developed algebraic solutions for 3-line algorithms, and Lowe [16] used Newton-Raphson method for any number of line segments. Liu, Huang and Faugeras combined [14] points and line segments into the same pose estimation procedure.

Motivated by the lack of methods which directly provide the unique solution for the redundant date case, in this paper, we will develop a family of linear algorithms for 4, 5 and n-point pose estimation in order to directly get the unique solution. We can therefore avoid using iterative approaches but make profit of the redundant data. Developing linear algorithms using redundant data for various vision tasks has always attracted attention of many researchers in computer vision [15, 19, 13, 17]. We establish that the pose is uniquely determined and can be linearly estimated with at least 4 correspondence points provided that the reference points together with the perspective center do not lie on the critical surfaces of configuration.

2 Review of nonlinear 3-point algorithm

As we have already mentioned in the introduction, the closed-form solution for 3 points has been known for a long time, and since then there have been many different approaches developed both in photogrammetry and computer vision. One of the most popular presentation of P3P in computer vision may be found in Fischler and Bolles [6], a detailed numerical analysis of the different algorithms was reported in Haralick et al. [9].

For n point correspondences $\mathbf{p}_i \leftrightarrow \mathbf{u}_i$ for $i = 1, \ldots, n$ between reference points $\mathbf{p}_i$ in space and image points $\mathbf{u}_i$, each pair of correspondences of points $\mathbf{p}_i \leftrightarrow \mathbf{u}_i$ and $\mathbf{p}_j \leftrightarrow \mathbf{u}_j$ gives the following constraint on the unknown distances $x_i = ||\mathbf{p}_i - \mathbf{c}||$ and $x_j = ||\mathbf{p}_j - \mathbf{c}||$ of the points $\mathbf{p}_i$ and $\mathbf{p}_j$ to the perspective center of the camera $\mathbf{c}$ (cf. Figure 2):

$$d_{ij}{}^2 = x_i{}^2 + x_j{}^2 - 2x_i x_j cos\theta_{ij},$$

where $d_{ij} = ||\mathbf{p}_i - \mathbf{p}_j||$ is the distance between i-th and j-th reference points and $\theta_{ij} = \angle \mathbf{u}_i \mathbf{c} \mathbf{u}_j$ the view angle from the perspective center of the camera between i-th and j-th image points.

This constraint is quadratic in the unknown depths, and can be rewritten as follows:

$$f_{ij}(x_i, x_j) = x_i{}^2 + x_j{}^2 - 2x_i x_j cos\theta_{ij} - d_{ij}{}^2 = 0. \quad (1)$$

For the case of $n = 3$, the following polynomial system is obtained

$$f_{12}(x_1, x_2) = 0,$$

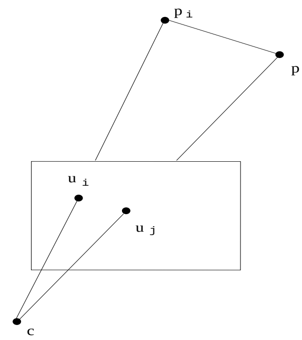

Figure 1: The basic geometry of the pose determination for each pair of the correspondences $\mathbf{p}_i \leftrightarrow \mathbf{u}_i$ and $\mathbf{p}_j \leftrightarrow \mathbf{u}_j$ between the reference points and the image points.

$$f_{13}(x_1, x_3) = 0,$$
$$f_{23}(x_2, x_3) = 0.$$

for three unknowns x_1, x_2, x_3. This system has Bezout bound of $8 = 2 \times 2 \times 2$ solutions. Since there is no linear terms in the equation, special solutions may be expected. Using the classical Sylvester resultant to eliminate x_3 between $f_{13}(x_1, x_3)$ and $f_{23}(x_2, x_3)$ to get a polynomial $h(x_1, x_2)$, then further eliminate x_2 between $f_{12}(x_1, x_2)$ and $h(x_1, x_2)$, a special 8-degree polynomial equation in one variable x_1 is obtained. This polynomial is in fact only a 4-degree polynomial equation in $x = x_1^2$:

$$g(x) = a_5 x^4 + a_4 x^3 + a_3 x^2 + a_2 x + a_1 = 0.$$

which has at most four solutions for x and can be solved in closed form. As x_i is positive, so $x_1 = \sqrt{x}$. Then x_2 and x_3 are uniquely determined from x_1.

Note that this 4-degree polynomial is different from that derived in [6].

3 The case of 4 points

Due to the multiplicity of the solutions of 3-point algorithm, in practice, we need a 4-th point or more for a unique solution, of course if the whole set of points together with the perspective center does not sit on the critical surfaces mentioned before.

For $n = 4$, an over-constraint system of 6 polynomial equations of type $f_{ij}(x_i, x_j) = 0$ is obtained for the 4 unknowns x_1, x_2, x_3, x_4. One straightforward approach is to take the subsets of 3 points of the set of 4 points, then solve the 4-degree polynomial equation for each subset, finally find the common solution. This is indeed the most frequent practice both in photogrammetry industry and computer vision. There are however several drawbacks to this approach. The first is that we have to solve several 4-degree polynomial equations. Secondly, we need to find the common solution of them which might be difficult due to noised data. Finally, probably the most important, we can not make profit of the redundant data besides of disambiguating the multiple solutions.

4 The linear 4-point algorithm

Our goal is to try to directly get the unique solution from the redundant polynomial equation system. In fact, finding the common roots is equivalent to the determination of the zero-dimensional variety generated by the ideal $(f_{12}, \ldots, f_{34})$. We can algebraically get a linear polynomial in generic cases and in one of the unknowns by successive application of Ritt method or pseudo division of polynomials over $I\!R(c_{ij}, d_{ij})[x_1, \ldots, x_i]$. This can effectively be done with any computer algebra system and will directly give the unique solution of the problem for general configurations of the points.

However this algebraic method is almost useless for practical numerical situations as the successive elimination will ultimately give complicated coefficients for the final linear polynomial which compromise the numerical stability of the solution. Instead of doing it algebraically, our goal is to develope a numerical linear method which indeed gives the unique solution if it does exist.

For n points, each pair of points gives a 4-degree polynomial equation, we can have $\frac{n(n-1)}{2}$ quadratic constraints of type $f_{ij}(x_i, x_j) = 0$ on the n unknown depthes $x_1, \ldots, x_n$, and $\frac{(n-1)(n-2)}{2}$ 4-degree polynomial equation of type $g(x) = 0$ in one variable $x = x_i^2$.

For $n = 4$, we obtain three 4-degree polynomial equations

$$\begin{aligned} g(x) &= a_5 x^4 + a_4 x^3 + a_3 x^2 + a_2 x + a_1 = 0 \\ g'(x) &= a'_5 x^4 + a'_4 x^3 + a'_3 x^2 + a'_2 x + a'_1 = 0 \\ g''(x) &= a''_5 x^4 + a''_4 x^3 + a''_3 x^2 + a''_2 x + a''_1 = 0. \end{aligned} \quad (2)$$

Let the 5-vector $\mathbf{t}_5 = (t_0, t_1, \ldots, t_4)^T = (1, x, \ldots, x^4)^T$, this can be rewritten in matrix form:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a'_1 & a'_2 & a'_3 & a'_4 & a'_5 \\ a''_1 & a''_2 & a''_3 & a''_4 & a''_5 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \mathbf{A}_{3\times5}\mathbf{t}_5 = 0.$$

This can be viewed as a homogeneous linear equation system in $t_i = x^i$ for $i = 0, \ldots, 4$. Since the matrix $\mathbf{A}_{3\times5}$ has at most rank $3 = \min(3, 5)$, the singular value decomposition of $\mathbf{A}_{3\times5}$ is

$$\mathbf{U}_{3\times5}\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, 0, 0)(\mathbf{v}_1, \ldots, \mathbf{v}_5)^T.$$

The null space of $\mathbf{A}_{3\times5}$ is spanned by the right singular vectors $\mathbf{v}_4$ and $\mathbf{v}_5$. Therefore we can construct a 2-dimensional solution space for $\mathbf{t}_5$, parameterized by λ and ρ:

$$\mathbf{t}_5 = \lambda\mathbf{v}_4 + \rho\mathbf{v}_5 \quad \text{for} \quad \lambda, \rho \in I\!R. \quad (3)$$

Now consider the nonlinear constraints among the components of $\mathbf{t}_5$, it can be easily checked that

$$t_i t_j = t_k t_l \Longleftrightarrow x^i x^j = x^k x^l \quad (4)$$
$$\text{for} \quad i + j = k + l, \quad 0 \le i, j, k, l \le 4.$$

The substitution of the 2-dimensional solution just obtained in (3) for t_i in the constraint (4) gives a homogeneous quadratic equation in λ and ρ:

$$b_1\lambda^2 + b_2\lambda\rho + b_3\rho^2 = 0,$$

where

$$\begin{aligned} b_1 &= \mathbf{v}_4^{(i)}\mathbf{v}_4^{(j)} - \mathbf{v}_4^{(k)}\mathbf{v}_4^{(l)}, \\ b_2 &= \mathbf{v}_4^{(i)}\mathbf{v}_5^{(j)} + \mathbf{v}_5^{(i)}\mathbf{v}_4^{(j)} - (\mathbf{v}_4^{(k)}\mathbf{v}_5^{(l)} + \mathbf{v}_5^{(k)}\mathbf{v}_4^{(l)}), \\ b_3 &= \mathbf{v}_5^{(i)}\mathbf{v}_5^{(j)} - \mathbf{v}_5^{(k)}\mathbf{v}_5^{(l)}. \end{aligned}$$

We can have 7 such equations for 7 different values of $\{(i, j, k, l), i+j = k+l \text{ and } 0 \le i, j, k, l \le 4\}$ modulo the possible permutations between i and j or k and l:

(i,j,k,l)
(4, 2, 3, 3)
(4, 1, 3, 2)
(4, 0, 3, 1)
(4, 0, 2, 2)
(3, 1, 2, 2)
(3, 0, 2, 2)
(2, 0, 1, 1)

These 7 quadratic equations can be stacked into the following matrix form if we denote $\mathbf{y}_3 = (y_0, y_1, y_2)^T = (\lambda^2, \lambda\rho, \rho^2)^T$:

$$\begin{pmatrix} b_1 & b_2 & b_3 \\ b'_1 & b'_2 & b'_3 \\ \vdots & \vdots & \vdots \\ b_1^{(6)} & b_2^{(6)} & b_3^{(6)} \end{pmatrix} \begin{pmatrix} \lambda^2 \\ \lambda\rho \\ \rho^2 \end{pmatrix} = \mathbf{B}_{7\times3}\mathbf{y}_3 = 0.$$

This overdetermined system of equations can be viewed as linear in λ^2, $\lambda\rho$ and ρ^2, and can be nicely solved by SVD decomposition as the right singular vector of the smallest singular value of $\mathbf{B}_{7\times3}$.

It is clear that these 7 equations are linearly independent but not algebraically. In fact three of such constraints are the necessary and sufficient algebraic conditions for x^i for $i = 0, \ldots, 4$ being an equal ratio series.

The 2-dimensional solution space defined by λ and ρ for $\mathbf{t}_5$ reduces now to one-dimensional by fixing the ratio of λ and ρ as

$$\lambda/\rho = y_0/y_1 \quad \text{or} \quad \lambda/\rho = y_1/y_2.$$

After obtaining the ratio λ/ρ, we can determine completely λ and ρ by using one of the scalar equation of the solution (3),

$$1 = \lambda\mathbf{v}_4^{(0)} + \rho\mathbf{v}_5^{(0)}.$$

Therefore $\mathbf{t}_5$ is completely determined. The final x is taken to be

$$x = t_1/t_0 \text{ or } t_2/t_1 \text{ or } t_3/t_2 \text{ or } t_4/t_3, \qquad (5)$$

or the average of all these values. Recall that $x = x_i^2$, the final positive depth $x_i = \sqrt{x}$.

We can therefore establish that *the pose of the calibrated camera is uniquely determined by 4 point correspondences provided that the 4 reference points together with the perspective center of the camera do not lie on the critical configurations. The unique solution can be estimated by the linear 4-point algorithm.*

It is important to note that the configuration of 4 coplanar reference points is not critical if they are not coplanar with the perspective center. It is known that the unique solution can be estimated linearly for a set of at least 4 coplanar points. The unique solution for 4 coplanar points can still be obtained by this algorithm.

5 The linear 5-point and n-point algorithms

From $n = 5$ on, it is interesting to see that there exist sufficient number of 4-degree polynomials for directly solving $t_i = x^i$ linearly, as $\frac{(n-1)(n-2)}{2} \geq \frac{(5-1)(5-2)}{2} = 6$.

For the case of $n = 5$, we can obtain 6 4-degree polynomials given in the following matrix form:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1' & a_2' & a_3' & a_4' & a_5' \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1^{(5)} & a_2^{(5)} & a_3^{(5)} & a_4^{(5)} & a_5^{(5)} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \mathbf{A}_{6\times 5}\mathbf{t}_5 = 0.$$

Let the singular value decomposition of $\mathbf{A}_{6\times 5}$ be $\mathbf{U}_{6\times 6}\Sigma_{6\times 5}\mathbf{V}_{5\times 5}^T$, the vector $\mathbf{t}_5$ is directly obtained as the right singular vector $\mathbf{v}_5$ of the smallest singular value of $\mathbf{A}_{6\times 5}$. Then x can be obtained in the same manner by Equation (5) as for the linear 4-point algorithm.

It is immediate that the same algorithm is also valid for any $n \geq 5$ points. We just need to SVD the $\frac{(n-1)(n-2)}{2} \times 5$ matrix $\mathbf{A}$ to get the solution for the vector $\mathbf{t}_5$.

We can therefore establish that *the pose of the calibrated camera is uniquely determined by $n \geq 5$ point correspondences provided these n points together with the perspective center of the camera do not lie on the critical configurations. The unique solution can be estimated by the linear n-point algorithm.*

6 Outline of the algorithms

The linear algorithms just described above can be outlined as follows:

- Preprocessing of the data:

  For each pair of n point correspondences $\mathbf{p}_i = (x_i, y_i, z_i)^T \Leftrightarrow \mathbf{u}_i = (u_i, v_i, 1)^T$, compute the distances of reference points d_{ij} and the viewing angle of image points (with the internal parameters of the camera) $\cos\theta_{ij}$.

- Solving the depths of the reference points:

  From the $(n-1)(n-2)/2$ 4-degree polynomial equations of form $g(x) = 0$, form the matrix $\mathbf{A}_{\frac{(n-1)(n-2)}{2}\times 5}$, then use SVD to linearly solve either in two steps for 4 points or in one step for more than 4 points for the vector $\mathbf{t}_5 = (1, x, \ldots, x^4)^T$. The square of the depth x is obtained by Equation (5), then the depth is set to $x_i = \sqrt{x}$.

- Absolute orientation:

  The recovered depths of the reference points give a complete estimation of the coordinates of the reference points in space $\tilde{\mathbf{p}}_i$ expressed actually in the camera-centered coordinate system. It remains the determination of a similarity transformation between two sets of 3D points $\tilde{\mathbf{p}}_i \Leftrightarrow \mathbf{p}_i$. The best rotation in the least squares sense can be found in closed form using the quaternion representation [11, 4]. The determination of the translation and the scale immediately follow from the rotation.

7 Experimental results

The two linear algorithms presented above for pose determination have been implemented and experimented both on simulated and real images.

First, the coordinates of 4 reference points in 3D are randomly generated within a cube of width 200 by a uniform random number generator. The orientation of the camera, represented by 3 Euler angles and the position of the camera are also randomly and uniformly generated. For each random pose of the camera, these randomly generated reference points are projected onto an image plane by a camera with a focal length 1500 pixels (without skew) and the principal point located at $(256, 256)$. Secondly, the positions of the image points are perturbed by a uniform noise. A noise of level σ means a uniform distribution in $[-\frac{\sigma}{2}, \frac{\sigma}{2}]$. Ten tests have been performed and the results of the linear 4-point and the linear 5-point algorithms are respectively shown in Table 1 and Table 2 in which the percentage of the relative error of the computed depth is given for each trial.

A special field of points is used for testing the pose estimation algorithms within CUMULI project. The images are taken by a Kodak DCS digital camera whose resolution is 1024×1536. Both the calibration of the camera and the camera pose are accurately determined by the standard photommetric technics. The results on 6 different positions of the camera and 28 points of the field are shown in Table 3 for the linear 4-point and in Table 4 for the linear 5-point algorithm.

For simulated data, we note that all linear algorithms perform very well, the relative errors are very small and degrade very gracefully with the increasing pixel errors. For the real data, the linear 4-point algorithm performs very well if indeed the good solution exsits for the configuration. However, there are several critical 4-point configurations where abnormal big errors occur (illustrated in bold in Table 3). For instance, in view 3 and test 3, the relative error reaches as high as 152%. This is a special configuration where two of the four points are very close in space, this reduces almost the 4 point configuration to a 3 point one, in addition, this 3 point configuration is still critical as it has a double solution. This has been verified by using the algebraic solutions of 3 subsets of 3 points of the 4 points. The ground truth for the depth of one point is 1.859, however the only subset that gives real solutions turns out 1.849 and 1.875 which are difficult to select the good one. In view 7 and test 4 (with 47% relative error), there is still double solutions in the configuration. The ground truth of one depth is 1.815, one subset has no real roots, the other two give $(1.808, 1.814)$ and $(1.807, 1.866)$. In view 9 and test 2 (with 46% relative error), the ground truth is 1.838. All 3 subsets give real solutions (1.345, 1.839), (1.315, 1.839), and (1.276, 1.838). This configuration is clearly critical since it has two feasible solutions one around 1.3 and another around 1.8. All other cases where the relative error exceeds 1% have the similar behaviours.

The critical configurations do not occur in our experiments for 5-point configurations, since the possibility that a set of randomly generated 5 points is critical is much smaller than that of a set of 4 points.

8 Discussion

The closed form solutions for the pose determination of the calibrated camera from 3 point correspondences has been known since long time. We presented in this paper a family of linear algorithms for n-point $(n \geq 4)$ pose determination. For 4 points, a two-step linear algorithm was developed and for $n \geq 5$ points, one-step linear algorithm was presented. These linear algorithms give the unique solution when the unique solution indeed exists, *i.e.* if the reference points are not sitting on the known critical surfaces. Our experimental results validated these algorithms.

The methodology used in this paper for pose determination can be easily applied to other problems in vision whereever the system of polynomial equations is overconstraint.

Acknowledgements

This work was partly supported by European project CUMULI which is gratfully acknowledged.

References

[1] H.H. Chen. In *Proceedings of the 3rd International Conference on Computer Vision, Osaka, Japan*, pages 374–378, 1990.

[2] D. Dementhon and L.S. Davis. *International Journal of Computer Vision*, 15(1/2):123–141, 1995.

[3] M. Dhome, M. Richetin, J.T. Lapresté, and G. Rives. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, 1989.

[4] O. Faugeras and M. Hebert. *The International Journal of Robotics Research*, 5:27–52, 1986.

[5] O.D. Faugeras and G. Toscani. In *Proceedings of International Workshop on Machine Vision and Machine Intelligence, Tokyo, Japan*, 1987.

[6] M.A. Fischler and R.C. Bolles. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.

[7] W. Förstner. *Computer Vision, Graphics and Image Processing*, 40:273–310, 1987.

[8] S. Ganapathy. In *Proc. of IEEE conference on Robotics and Automation*, pages 130–139, 1984.

[9] R.M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA*, pages 592–598, 1991.

[10] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle. *Computer Vision, Graphics and Image Processing*, 47:33–44, 1989.

[11] B.K.P. Horn. *Journal of the Optical Society of America*, 5(7):1127–1135, 1987.

[12] R.K. Lenz and R.Y. Tsai. In *Proceedings of* IEEE *International Conference on Robotics and Automation, Raleigh, North Carolina, USA*, pages 68–75, 1987.

[13] Y. Liu and T.S. Huang. *Computer Vision, Graphics and Image Processing*, 44(1):35–57, October 1988.

[14] Y. Liu, T.S. Huang, and O.D. Faugeras. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, January 1990.

[15] H.C. Longuet-Higgins. In *Nature*, volume 293, pages 133–135. XX, September 1981.

[16] D.G. Lowe. kluwer Academic Publishers, 1985.

[17] J. Philip. *Photogrammetric Record*, 15(88):589–599, October 1996.

[18] C.C. Slama, editor. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[19] M. Spetsakis and J. Aloimonos. *International Journal of Computer Vision*, 4:171–183, 1990.

[20] I.E. Sutherland. *Proceedings of the IEEE*, 62:453–461, 1974.

[21] E.H. Thompson. *Photogrammetric Record*, X(27):201–204, 1966.

[22] B.P. Wrobel. In *Calibration and Orientation of Cameras in Computer Vision.*, Washington DC, August 1992. Springer-Verlag.

[23] J.S.C. Yuan. and orientation. IEEE *Transactions on Robotics and Automation*, 5(2):129–142, 1989.

Noise Level σ	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Test 1	0.1	0.002	0.03	0.2	0.1	0.5	0.2	0.8	0.3	0.06
Test 2	0.002	0.001	0.05	0.004	0.09	0.1	0.05	0.05	0.1	0.2
Test 3	0.02	0.00009	0.04	0.02	0.09	0.1	0.2	0.3	0.04	0.1
Test 4	0.03	0.02	0.05	0.08	0.02	0.2	0.2	0.06	0.5	0.2
Test 5	0.01	0.1	0.1	0.1	0.1	0.2	0.1	0.2	0.2	0.1
Test 6	0.02	0.01	0.03	0.1	0.2	0.1	0.2	0.02	0.02	0.2
Test 7	0.3	0.2	0.5	0.5	0.8	0.09	0.4	2.5	1	1.5
Test 8	0.004	0.02	0.06	0.2	0.1	0.08	0.06	0.07	0.08	0.03
Test 9	0.001	0.02	0.02	0.004	0.04	0.04	0.03	0.01	0.01	0.06
Test 10	0.3	1.0	1.0	0.3	0.2	2.2	0.7	5.0	2.6	2.4
median	0.02	0.02	0.05	0.1	0.1	0.1	0.2	0.185	0.15	0.15

Table 1: The relative error in percentage of the computed depth by the linear 4-point algorithm with the uniform pixel errors.

Noise Level σ	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Test 1	0.02	0.06	0.4	0.5	0.4	0.2	0.5	0.7	1.1	0.6
Test 2	0.2	0.1	0.5	0.3	0.7	0.7	1.1	0.2	1.0	2.0
Test 3	0.06	0.03	0.03	0.03	0.05	0.2	0.2	0.2	0.03	0.3
Test 4	0.01	0.002	0.04	0.07	0.04	0.1	0.07	0.2	0.2	0.2
Test 5	0.001	0.01	0.03	0.06	0.05	0.03	0.02	0.09	0.2	0.08
Test 6	0.002	0.1	0.2	0.8	0.2	0.03	0.1	0.1	0.1	0.6
Test 7	0.02	0.04	0.2	0.2	0.03	0.1	0.3	0.2	0.3	0.6
Test 8	0.08	0.02	0.1	0.05	0.1	0.2	0.2	0.01	0.3	0.8
Test 9	0.005	0.02	0.03	0.001	0.05	0.03	0.01	0.08	0.01	0.03
Test 10	0.1	0.1	0.1	0.6	0.3	0.2	0.6	0.4	2.2	1.8
median	0.02	0.035	0.1	0.135	0.075	0.15	0.2	0.15	0.25	0.6

Table 2: The relative error in percentage of the computed depth by the linear 5-point algorithm with uniform pixel errors.

# test	1	2	3	4	5	6	7	8	9	10	median
pos. 1	0.04	0.02	0.04	0.03	0.2	**2**	0.005	0.03	0.01	0.06	0.03
pos. 3	0.2	0.03	**152**	**2**	0.4	**2**	0.02	0.02	**1**	0.02	0.3
pos. 5	0.01	0.1	0.04	0.01	0.02	0.09	0.004	0.2	0.04	0.1	0.04
pos. 7	0.09	0.06	0.08	**47**	0.003	0.05	0.03	0.3	0.1	0.09	0.09
pos. 8	0.04	0.01	0.005	0.03	0.003	**2**	0.09	0.01	0.02	0.04	0.03
pos. 9	0.004	**46**	0.05	0.2	0.04	0.8	0.02	0.3	0.05	0.02	0.05
median	0.04	0.04	0.05	0.1	0.03	**1**	0.02	0.1	0.04	0.05	

Table 3: The relative error in percentage of the computed depth by the linear 4-point algorithm with the real image data.

# test	1	2	3	4	5	6	7	8	9	10	median
pos. 1	0.2	0.2	0.6	0.2	0.2	1.7	0.2	0.02	0.1	1.8	0.2
pos. 3	0.04	0.4	0.03	0.5	0.07	0.2	0.1	0.3	0.04	0.01	0.085
pos. 5	0.02	0.09	0.004	0.09	0.3	0.01	0.04	0.0002	0.03	0.004	0.025
pos. 7	0.009	0.02	0.02	0.02	0.2	0.3	0.08	0.05	0.02	0.05	0.035
pos. 8	0.008	0.9	0.2	0.2	0.3	0.05	0.02	0.07	0.7	0.2	0.2
pos. 9	0.09	0.4	0.2	0.06	0.05	0.03	0.006	0.1	0.07	0.07	0.07
median	0.03	0.3	0.115	0.145	0.3	0.175	0.06	0.06	0.055	0.06	

Table 4: The relative error in percentage of the computed depth by the linear 5-point algorithm with the real image data.