Iterative improvement of motion parameters obtained by a linear algorithm.

by Johan Philip

Abstract::The motion parameters obtained from noisy data by a linear algorithm using six or more points do often suffer from low accuracy. We describe an iterative method, which in general doubles the accuracy.

1. Introduction

We consider the determination of the motion of a rigid object in 3D from 2D perspective images of it. We assume that a number of points on the object have been identified and matched in two images. We shall call two match points a *point pair* and shall discuss how the motion of the object from one image to the other can be determined. Theoretically, there are five motion parameters to be determined, so at least five point pairs are needed. With five point pairs, the solution is not unique. For six or more point pairs, there exists algorithms that give a unique solution for exact data except in some degenerate cases. Even for moderately noisy data, many more than six points are needed to get a reasonable result, Weng, Huang and Ahuja (1988), Philip (1991).

A linear method for six point pairs was published by Hofmann-Wellenhof (1979). An improved version of this algorithm together with a proof of the uniqueness was given by Philip (1996). Linear methods for eight point pairs based on a different idea were published by Stefanovic (1973) and Tsai and Huang (1984).

All these algorithms consist of two consecutive optimization steps which makes the solution suboptimal. In this paper we shall describe a way of obtaining the optimal solution in a sense specified below.

We introduce some notation and give the mathematical formulation in Secs. 2 and 3. The iterative method is described in Sec. 5. Some computational experience is reported in Sec.6 and discussed in Sec. 7.

2. Notations, mathematical description and the essential matrix.

The description of this section assumes that the data are exact and that there exists an object in 3D and a motion giving these data.

We consider a pinhole camera and denote the 3D coordinates of point i before and after the motion by $\boldsymbol{U^i}$ and $\boldsymbol{V^i}$, respectively. These coordinates are not observed, but only the perspective coordinates

$$\boldsymbol{u}^i = \begin{bmatrix} U^i{}_1/U^i{}_3 \\ U^i{}_2/U^i{}_3 \\ 1 \end{bmatrix} \qquad \boldsymbol{v}^i = \begin{bmatrix} V^i{}_1/V^i{}_3 \\ V^i{}_2/V^i{}_3 \\ 1 \end{bmatrix}. \tag{2.1}$$

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

$$\boldsymbol{V}^i = \boldsymbol{R}\boldsymbol{U}^i + \boldsymbol{t} \tag{2.2}$$

Since the distance to the object is not known, the length of $\boldsymbol{t}$ cannot be determined so the sought motion parameters are the three parameters describing $\boldsymbol{R}$ and the two parameters describing the direction of $\boldsymbol{t}$. By (2.2), the vectors $\boldsymbol{V}^i$, $\boldsymbol{R}\boldsymbol{U}^i$, and $\boldsymbol{t}$ are in the same plane and are said to satisfy the *coplanarity condition*. This implies that the determinant formed of the vectors is zero. This can also be written

$$\boldsymbol{V}^{i^t}\boldsymbol{C}\boldsymbol{R}\boldsymbol{U}^i = 0, \tag{2.3}$$

where

$$\boldsymbol{C} = \begin{bmatrix} 0 & t_3 & -t_2 \\ -t_3 & 0 & t_1 \\ t_2 & -t_1 & 0 \end{bmatrix} \tag{2.4}$$

is the matrix performing cross product with $\boldsymbol{t}$. The *essential matrix* $\boldsymbol{E}$ is defined by

$$\boldsymbol{E} = \boldsymbol{C}\boldsymbol{R}. \tag{2.5}$$

After division by $U^i{}_3 V^i{}_3$ in (2.3), we obtain the following homogeneous equation for $\boldsymbol{E}$

$$\boldsymbol{v}^{i^t}\boldsymbol{E}\boldsymbol{u}^i = 0 \tag{2.6}$$

from each point pair. Since $\boldsymbol{E}$ can only be determined up to a multiplicative constant, one of its elements can be chosen arbitrarily, e.g. equal to one. The methods for eight point pairs use eight equations of type (2.6) to determine the remaining eight elements of $\boldsymbol{E}$. The methods for six point pairs use six equations of type (2.6) and the fact that $\boldsymbol{E}$ is not an arbitrary 3×3-matrix but is of the form $\boldsymbol{C}\boldsymbol{R}$. When $\boldsymbol{E}$ is found, two $\boldsymbol{R}$ and two $\boldsymbol{C}$ satisfying (2.5) can be determined, [6]. A unique set of motion parameters is found by using the fact that the object is in front of the camera both before and after the motion, i.e. $U^i{}_3 > 0, \quad V^i{}_3 > 0$.

3. Noisy data.

We still assume that there exists an object in 3D which has been moved. The mesurement errors show up as noise in the data $\boldsymbol{u}^i$ and $\boldsymbol{v}^i$. Because of the noise, equation (2.6) will not hold for the true $\boldsymbol{E}$ but we have

$$(\boldsymbol{v}^i)^t \boldsymbol{E}\boldsymbol{u}^i = y_i \tag{3.1}$$

where y_i is called *the deviation from coplanarity* for point pair i . If one has exactly six or eight point pairs, one can get an estimate $\tilde{\boldsymbol{E}}$ of $\boldsymbol{E}$ by putting $y_i = 0$ in all the equations (3.1). A more common situation is that one has $N > 8$ point pairs and solves the least squares problem

$$\min_{\boldsymbol{E}} \sum_{i=1}^{N} w_i y_i^2 \tag{3.2}$$

where $w_i > 0$ are some weights and $\boldsymbol{E}$ is assumed to be *any* 3×3-matrix, though normalized e.g. by requiring $e_{33} = 1$. Solving (3.2) is often the first linear step of a many point algorithm and results in an estimate $\tilde{\boldsymbol{E}}$, which cannot be written as the product of a skew symmetric matrix and a rotation matrix as in (2.5). The second step is to find $\boldsymbol{C}$ and $\boldsymbol{R}$ such that $\boldsymbol{CR}$ approximates $\tilde{\boldsymbol{E}}$, see [3]. Solving (3.2) means projecting from an N-dimensional space to a linear 8-dimensional subspace. The second step is a projection from the 8-dimensinal subspace on a 'nonlinear' 5-dimensional subset. This two step procedure is not optimal and it is the purpose of this paper to describe how to compute the projection directly from the N-dimensional data space to the motion parameter set, which is descibed by five parameters. We do this by substituting (3.1) in (3.2) and obtain the problem

$$\min_{\boldsymbol{C},\boldsymbol{R}} \sum_{i=1}^{N} w_i \left((\boldsymbol{v}^i)^t \boldsymbol{C}\boldsymbol{R}\boldsymbol{u}^i \right)^2 , \tag{3.3}$$

where $\boldsymbol{C}$ is normalized by $\|\boldsymbol{t}\| = 1$. This formula can be used for $N \geq 5$.
There are various strategies for handling noisy data depending on the character of the noise. If all $\boldsymbol{u}^i$ and $\boldsymbol{v}^i$ have a small error, one can start my solving (3.2) and use the obtained $\tilde{\boldsymbol{E}}$ as starting point for an iterative solution of (3.3). If some $\boldsymbol{u}^i$ and $\boldsymbol{v}^i$ are outliers meaning that they have much larger errors than the others, probably because of mismatching, it can be a good strategy to pick a subset of six or eight point pairs and solve for $\tilde{\boldsymbol{E}}$. Then, y_i is calculated for the remaining data with this $\tilde{\boldsymbol{E}}$ and this process is repeated for several choices of the subset. The choice resulting in the smallest y_i for most of the remaining data, can be used as starting point for (3.3), see [1].

4. The weights.

Any external knowledge about the accuracy of separate measurements can be used for choosing the weights different from $w_i = 1$ for all i. We shall describe one way of computing the weights from the obtained projective coordinates even though the numerical tests indicate that the obtained accuracies are somewhat poorer than with $w_i = 1$ for all i.

When the least squares problem (3.2) has been solved for $N > 8$, one has an estimate of the variance of the residual $r_i = y_i - 0$, which can be used to choose w_i for problem (3.3). If (3.2) is solved by computing the singular value decomposition (SVD) of the system matrix, the residual variances are easily acquired. Denoting the system matrix by A and letting its SVD be $A = USV^t$, the so called hat matrix H is

$$H = A(A^tA)^{-1}A^t = UDU^t, \tag{4.1}$$

where D is a diagonal matrix with $d_{ii} = 1$ for $1 \le i \le 8$ and $d_{ii} = 0$ for $8 < i \le N$. The variance of r_i is proportional to $1 - h_{ii}$ and one can choose

$$w_i = \frac{1}{1 - h_{ii}}. \tag{4.2}$$

5. Iterative improvement.

We shall describe a technique for solving the nonlinear problem (3.3). For the iterations to converge to the correct solution, it is imperative to have a starting point close to this solution.

A rotation matrix $\boldsymbol{R}$ is an orthogonal 3×3-matrix with determinant = $+1$. There are various ways of representing such a matrix as a function of three parameters. We have tested several representations and chosen the following because of its symmetry, lack of bounds or conditions on the parameters and its amenability to differentiation.

Let the rotation angle be θ and the rotation axis with length one be $\hat{\boldsymbol{a}} = (a_1, a_2, a_3)$. Define $\boldsymbol{n} = \tan(\theta/2)\hat{\boldsymbol{a}}$ and let $\|\boldsymbol{n}\|$ be its l_2-norm.

With the notations $diag = (1 - \|\boldsymbol{n}\|^2)/2$ and $denom = (1 + \|\boldsymbol{n}\|^2)/2$ we have

$$\boldsymbol{R} = \frac{1}{denom}\begin{bmatrix} n_1^2 + diag & n_1n_2 - n_3 & n_1n_3 + n_2 \\ n_2n_1 + n_3 & n_2^2 + diag & n_2n_3 - n_1 \\ n_3n_1 - n_2 & n_3n_2 + n_1 & n_3^2 + diag \end{bmatrix}. \tag{5.1}$$

We avoid the side condition $\|\boldsymbol{t}\| = 1$ in the minimization by letting $\boldsymbol{t} = (t_1, t_2, t_3)$ have arbitrary length and reformulate (3.3) as

$$min\phi = \min_{\boldsymbol{n},\, \boldsymbol{t}} \frac{1}{||\boldsymbol{t}||^2} \sum_{i=1}^{N} w_i \left((\boldsymbol{v}^i)^t \boldsymbol{CRu}^i \right)^2 . \qquad (5.2)$$

One can solve (5.2) by alternately minimizing with respect to $\boldsymbol{n}$ and $\boldsymbol{t}$. This seems to be advantageous because the minimization with respect to $\boldsymbol{t}$ is a linear problem. We tried this but it turned out to be very slow. We take, however, one initial step by minimizing (5.2) with respect to $\boldsymbol{t}$. Then it is suitable to write the minimization problem in the form

$$\min_{||\boldsymbol{t}|| = 1} \sum_{i=1}^{N} w_i \left(\boldsymbol{v}^i \times \boldsymbol{Ru}^i \cdot \boldsymbol{t} \right)^2 \qquad (5.3)$$

Introduce the vectors $\boldsymbol{z}_i = \sqrt{w_i}\boldsymbol{v}^i \times \boldsymbol{Ru}^i$ and form an $N \times 3$-matrix $\boldsymbol{Z}$ with rows equal to the $\boldsymbol{z}_i$. Then, (5.3) is equivalent to

$$\min_{||\boldsymbol{t}|| = 1} ||\boldsymbol{Zt}|| . \qquad (5.4)$$

Let the singular value decomposition of $\boldsymbol{Z}$ be $\boldsymbol{Z} = \boldsymbol{HSK}^t$ where $\boldsymbol{H}$ and $\boldsymbol{K}$ are orthogonal and $\boldsymbol{S}$ is diagonal with the nonincreasing singular values s_1, s_2 and s_3 along the diagonal. Then, the minimum in (5.4) equals the smallest singular value = s_3 and it is attained for $\boldsymbol{t}$ equal to $\pm$ the corresponding right singular vector, which makes up the third column of $\boldsymbol{K}$. As starting value for the iteration we use this $\boldsymbol{t}$ and chose its sign so that it best approximates the translation obtained in step one.
We use the conjugate gradient method for solving (5.2). The formulas for the gradients with respect to $\boldsymbol{n}$ and $\boldsymbol{t}$ are given in the Appendix. It is important to iterate in all six parameters at the same time. If one tries separate steps in $\boldsymbol{n}$ and $\boldsymbol{t}$, these steps become very short and the convergence becomes very slow.

6. Numerical tests.

The iterative improvement has been tested both on real photogrammetry data and on artificial data. In photogrammetry, the matched points often sit approximately in a plane, which makes the equations to be solved almost degenerate. With randomly generated artificial point pairs, the problem is seldom degenerate.
We present in Table 1 the results from tests with random objects consisting of N point pairs, random motions and random noise.
The rotation angle θ was generated in the interval (0,30) degrees while the $\boldsymbol{n}$- and $\boldsymbol{t}$-

directions were evenly distributed over all directions.
The error measures are all in degrees and they are :
$\Delta\theta$ = error in rotation angle
$\Delta\boldsymbol{n}$ = angle between true and calculated rotation axis
$\Delta\boldsymbol{t}$ = angle between true and calculated translation vector.
These errors stem from the noise added to the projective coordinates. On the average, the ratio between noise addition and projective coordinate is 1/1000.
The weights are $w_i = 1$.
The numbers presented in the table are *medians* of the obtained errors. We prefer medians to averages since the calculated motion parameters have error distributions that are far from symmetric.
The ratio between numbers in the same position in the first and second column gives the improvement achieved by the iterative method.
The numbers marked “sorted” are from a subset of all tests otained by sorting out tests suspected to have bad accuracy with the aid of test variables as described in [3]. Every number in the columns marked “all” is based on 1000 tests and “#data” gives the number of data remaining after the sorting. The sorting removes most of the bad estimates.
On the average, ten iterations are needed for the iterative improvement which increases the computation time by a factor 2.5 .

		from linear method			with iterative improvment		
		all	sorted	#data	all	sorted	#data
	$\Delta\theta$	1.3	.73		.77	.40	
$N = 6$	$\Delta\boldsymbol{n}$	12	5.98	740	7.59	3.4	740
	$\Delta\boldsymbol{t}$	2.2	1.17		1.22	.53	
	$\Delta\theta$	.60	.43		.20	.14	
$N = 8$	$\Delta\boldsymbol{n}$	5.1	3.2	640	1.7	1.1	630
	$\Delta\boldsymbol{t}$	1.0	.65		.28	.19	
	$\Delta\theta$	.17	.13		.079	.066	
$N = 12$	$\Delta\boldsymbol{n}$	1.4	.84	676	.67	.49	679
	$\Delta\boldsymbol{t}$	.30	.21		.13	.10	
	$\Delta\theta$	.088	.065		.055	.044	
$N = 20$	$\Delta\boldsymbol{n}$	.65	.45	696	.43	.35	713
	$\Delta\boldsymbol{t}$	.16	.12		.089	.075	

TABLE 1. Medians of errors. Each number in the "all" columns is based 1000 tests.

7. Discussion

As is seen in the table, the improvement varies with the number of point pairs N. From $N = 8$ and upwards the eight point algorithm is used and the resulting accuracy increases with N so there is less to be gained by the iterative improvement. For $N = 6$, we use the six point algorithm of [4] which produces an essential matrix that is the product of a skew symmetric matrix and a rotation matrix. Theoretically, five point pairs are needed for this but six noisy point pairs are used so the the algorithm produces some kind of estimate of the motion parameters. Generally, this estimate doesn't minimize the objective in (3.3) but it can be further decreased by the iterative improvement. Since the linear method is 'better' for $N = 6$, the gain is smaller than with $N = 8$.

One reason for $\Delta\boldsymbol{n}$ to be much bigger than $\Delta\theta$ and $\Delta\boldsymbol{t}$ is that the rotation axis is very uncertain for small rotation angles θ.

8. Appendix

The conjugate gradient method requires formulas for the partial derivatives of the objective funtcion ϕ with respect to the variables t_k and n_k for $1 \le k \le 3$. The $\boldsymbol{t}$-derivatives are obtained by writing

$$\phi = \frac{1}{\|\boldsymbol{t}\|^2} \sum_{i=1}^{N} w_i y_i^2 = \frac{1}{\|\boldsymbol{t}\|^2} \sum_{i=1}^{N} w_i (\boldsymbol{v}^i \times \boldsymbol{R}\boldsymbol{u}^i \cdot \boldsymbol{t})^2 . \qquad (8.1)$$

We get

$$\frac{1}{2}\frac{\partial \phi}{\partial t_k} = \frac{-t_k}{\|\boldsymbol{t}\|^4} \sum_{i=1}^{N} w_i y_i^2 + \frac{1}{\|\boldsymbol{t}\|^2} \sum_{i=1}^{N} w_i y_i \frac{\partial y_i}{\partial t_k} = \frac{1}{\|\boldsymbol{t}\|^2} \sum_{i=1}^{N} w_i y_i \left(\frac{\partial y_i}{\partial t_k} - \frac{y_i t_k}{\|\boldsymbol{t}\|^2} \right) , \qquad (8.2)$$

where

$$\frac{\partial y_i}{\partial t_k} = (\boldsymbol{v}^i \times \boldsymbol{R}\boldsymbol{u}^i)_k . \qquad (8.3)$$

To get the $\boldsymbol{n}$-derivatives, we rewrite y_i as

$$y_i = (\boldsymbol{v}^i)^t \boldsymbol{C}\boldsymbol{R}\boldsymbol{u}^i = (\boldsymbol{t} \times \boldsymbol{v}^i) \cdot \boldsymbol{R}\boldsymbol{u}^i . \qquad (8.4)$$

We get

$$\frac{1}{2}\frac{\partial \phi}{\partial n_k} = \frac{1}{\|\boldsymbol{t}\|^2} \sum_{i=1}^{N} w_i y_i (\boldsymbol{t} \times \boldsymbol{v}^i) \cdot \frac{\partial \boldsymbol{R}}{\partial n_k} \boldsymbol{u}^i , \qquad (8.5)$$

where $\frac{\partial \boldsymbol{R}}{\partial n_k}$ is obtained by differentiation of (5.1) . We give the expression for one index

$$\frac{\partial \boldsymbol{R}}{\partial n_1} = \frac{1}{denom} \left\{ \begin{bmatrix} n_1 & n_2 & n_3 \\ n_2 & -n_1 & -1 \\ n_3 & 1 & -n_1 \end{bmatrix} - \boldsymbol{R} \right\} . \qquad (8.6)$$

9. References

[1] P. Axelsson, “Autonomous Decisions in Photogrammetry using Minimum Description Length”, TRITA-GEOFOTO Report 1996:11, ISSN 1104-1145.

[2] Hofmann-Wellenhof, “Die gegenseitige Orientierung von zwei Strahlenbuendeln bei Uebereinstimmung, bei unbekannten Näherungswerten und durch ein nichtiteratives Verfahren”, Deutsche Geodätische Kommision, Reihe C, Nr. 257, 1979.

[3]. J. Philip, “Estimation of Three-Dimensional Motion of Rigid Objects from Noisy Observations”, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-13, January 1991, pp. 61-66.

[4] J. Philip, “A noniterative algorithm for determining all essential matrices corresponding to five point pairs”, Photogrammetric Record, 15(88), October 1996, pp.589-599.

[5] P. Stefanovic, “Relative Orientation - a new approach”, ITC-Journal, vol. 3, pp. 417-448, 1973.

[6] R.Y. Tsai and T.S. Huang, “Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces”, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-6, Jan. 1984.

[7] J. Weng, N. Ahuja, and T.S. Huang, “Closed-form solution + maximum likelihood: A robust approach to motion and structure estimation”, Proc. IEEE Conf. Comput. Vision Patt. Recogn., (Ann Arbor MI), June 5-9, 1988, pp. 381-386.