

# The Role of Total Least Squares in Motion Analysis

Matthias Mühlich and Rudolf Mester

Inst. f. Applied Physics, J. W. Goethe-Universität Frankfurt/Main, Germany  
(muehlich|mester)@iap.uni-frankfurt.de

**Abstract.** The main goal of this paper is to put well-established techniques for two-view motion analysis in the context of the theory of Total Least Squares and to make clear that robust and reliable motion analysis algorithms cannot be designed without a thorough statistical consideration of the consequences of errors in the input data.

We focus on the non-iterative 8+n-point algorithm for estimating the fundamental matrix and present a comprehensive statistical derivation of the compelling necessity for one of the normalization transforms proposed by Hartley [1, 2]. It turns out that without these transformations the results of the well-known non-iterative methods for two-view motion analysis are biased and inconsistent. With some further improvements proposed in this paper, the quality of the algorithm can even be enhanced beyond what has been reported in the literature before.

## 1 Introduction

The computation of relative orientation is one of the key points in two-view motion analysis. The early investigations of Longuet-Higgins [3] and Tsai & Huang [4] have led to a family of non-iterative<sup>1</sup> algorithms that have attracted considerable attention due to their moderate computational effort. Unfortunately, the quality of the results that are obtained from these algorithms decreases dramatically as soon as measurement errors enter the input data, and even the usage of far more data than is minimally required – the approved recipe: ‘redundancy combats noise’ – does not significantly defuse this situation.

We will consider in the following the non-iterative algorithm for the determination of the fundamental matrix as it is rather compactly described in [2], and which goes back to [4] (although the early papers are dealing with the case of calibrated cameras and employ the *essential matrix* which is a special case of the fundamental matrix).

---

<sup>1</sup> These algorithms are non-iterative except for the solution of eigensystem equations or the determination of the singular value decomposition which play a key role in these procedures. Sometimes these algorithms are called linear, but this is rather misleading since the computation of the eigensystem and the SVD is definitely non-linear.

## 2 The ideal case: no measurement error in the input data

Let us first briefly review the 8+n-point algorithm. Let  $\mathbf{u} = (u_1, u_2, 1)^T$  and  $\mathbf{v} = (v_1, v_2, 1)^T$  be the homogeneous coordinates of the projection of the same rigid object point in two images  $B_u$  and  $B_v$ . If the vectors  $\mathbf{u}$  and  $\mathbf{v}$  are error-free, it is well known that they are related to each other by the equation

$$\mathbf{v}^T \cdot \mathbf{F} \cdot \mathbf{u} = 0 \quad (1)$$

with a  $3 \times 3$ -matrix  $\mathbf{F}$ , the *fundamental matrix*, which is identical for all pairs of corresponding vectors  $\mathbf{u}_i, \mathbf{v}_i$ . The matrix  $\mathbf{F} = (f_{ij})$  is the entity we are interested in, since it encapsulates all the information on motion or relative orientation that can be extracted from the given image pair. The matrix  $\mathbf{F}$  is necessarily rank-deficient, i.e.  $\text{rank}(\mathbf{F}) = 2$  [4, 5]. By ‘vectorizing’ the matrix  $\mathbf{F}$ , i. e. stacking the matrix elements  $f_{ij}$  to form a vector  $\mathbf{f}$  which is (dropping double indices)

$$\mathbf{F} \rightarrow \mathbf{f} = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T = (f_1, \dots, f_9)^T,$$

equation (1) can be expressed as

$$\mathbf{a}^T \cdot \mathbf{f} = 0 \quad \text{with} \quad \mathbf{a}^T = (v_1 u_1, v_1 u_2, v_1, v_2 u_1, v_2 u_2, v_2, u_1, u_2, 1). \quad (2)$$

Each point correspondence ( $\mathbf{u}_i \leftrightarrow \mathbf{v}_i$ ) can be expressed in the form of equation (2) and from  $N$  correspondences we obtain a linear equation system

$$\mathbf{A}_0 \cdot \mathbf{f} = 0 \quad (3)$$

where the coefficient vectors  $\mathbf{a}_i^T$  ( $i = 1, \dots, N$ ) are row vectors of a  $N \times 9$ -matrix  $\mathbf{A}_0$  (subscript 0 denoting the unperturbed matrix).

If the correspondences between each pair  $\mathbf{u}_i, \mathbf{v}_i$  were perfect, matrix  $\mathbf{A}_0$  would have rank 8 for  $N \geq 8$  except for degenerate configurations<sup>2</sup> of the object points in the 3D-space [6], so 8 point correspondences would be sufficient for determining vector  $\mathbf{f}$  and thus the fundamental matrix  $\mathbf{F}$  as well. In this error-free case, equation (3) can be solved exactly, and the solution vector (or vectors, as in the case of degenerate configurations) correspond to the basis vector(s) of the nullspace of matrix  $\mathbf{A}_0$ .

## 3 The realistic case: input data with measurement errors

In the presence of errors, the matrix will not be of rank 8, i.e. there is no non-trivial solution to eqn.(2). Traditionally, the approach taken in this situation is as follows:

<sup>2</sup> It is not true that in the case of degenerate configurations no solution can be extracted at all; instead, we obtain a linear manifold of solution vectors, which could possibly be valuable information as well.

**Step 1: initial estimate of  $\mathbf{F}$ .** Since it follows from  $\text{rank}(\mathbf{A}) = 9$  that there is no  $\mathbf{f} \neq \mathbf{0}$  that solves eqn.(3) exactly, we try to find a unit vector  $\mathbf{f}$  for which the right hand side is at least close to  $\mathbf{0}$ . This means that we minimize  $|\mathbf{A}\mathbf{f}|^2$  subject to the constraint  $|\mathbf{f}|^2 = \mathbf{f}^T \mathbf{f} = 1$ . The solution to this constrained minimization problem obtained by means of Lagrangian multipliers is given by the eigenvector of  $\mathbf{A}^T \mathbf{A}$  corresponding to the smallest eigenvalue of  $\mathbf{A}^T \mathbf{A}$ . In other words (see [7]),  $\mathbf{f}$  is the right hand singular vector corresponding to the smallest singular value of matrix  $\mathbf{A}$ .

**Step 2: enforcement of  $\text{rank}(\mathbf{F}) = 2$ .** Since the true fundamental matrix  $\mathbf{F}$  must have rank 2, the vector  $\mathbf{f}$  obtained so far is rearranged to matrix form and the resulting matrix is enforced to have rank 2 by expanding the current estimate of  $\mathbf{F}$  in a sum of rank 1 matrices and suppressing the matrix with the lowest Frobenius norm. Practically, this is done by means of the *singular value decomposition* (SVD).

This procedure is correct under certain conditions that we will discuss later. However, it blocks the view onto a much more general framework which allows a consideration of the detailed statistical structure of the disturbances.

### 3.1 Error models and metrics for optimum estimation

Let us look a bit closer onto equation (3). It necessarily holds for error free data, i. e. for vector pairs ( $\mathbf{u}_i \leftrightarrow \mathbf{v}_i$ ) containing no measurement errors and, consequently, for row vectors  $\mathbf{a}_i^T$  being numerically correct. However, in practice there *are* errors in our input data and we have the following situation:  $\mathbf{A}\mathbf{f} \neq \mathbf{0}$  due to the fact that the matrix  $\mathbf{A}$  we actually have is related to the true, but unknown matrix  $\mathbf{A}_0$  by  $\mathbf{A} = \mathbf{A}_0 + \mathbf{D}$ , introducing the error matrix  $\mathbf{D} \neq \mathbf{0}$ .

$$(\mathbf{A}_0 + \mathbf{D})\mathbf{f} = \mathbf{A}_0\mathbf{f} + \mathbf{D}\mathbf{f} = \mathbf{D}\mathbf{f} \neq \mathbf{0}$$

In this situation the search for the solution vector  $\mathbf{f}$  boils down to *estimate* the true matrix  $\mathbf{A}_0$ , and given its corrupted version  $\mathbf{A}$  this is equivalent to estimate the error matrix  $\mathbf{D}$ . In other words, given a rank 9 matrix  $\mathbf{A}$  we have to find a plausible (whatever that may denote) estimate of a matrix  $\mathbf{D}$  that lowers the rank of  $\mathbf{A} - \mathbf{D} = \hat{\mathbf{A}}_0$  to 8, since for  $N \times 9$ -matrices  $\hat{\mathbf{A}}_0$  of rank  $\leq 8$  the equation  $\hat{\mathbf{A}}_0\mathbf{f} = \mathbf{0}$ ,  $\mathbf{f} \neq \mathbf{0}$  has one (or several) solution(s). The credibility of the estimate  $\mathbf{D}$  is inevitably related to a consideration of the structure of the stochastic process that generates the error matrix  $\mathbf{D}$ .

Let us assume for the moment (until we still look a bit closer) that the error matrix  $\mathbf{D}$  is a realization of a random matrix process with the following characteristics:

#### Model 1 (Zero-mean i.i.d. error matrix $\mathbf{D}$ )

$$\begin{aligned} \mathbb{E}[\mathbf{D}] &= \mathbf{0} \quad \Leftrightarrow \quad \mathbb{E}[d_{ij}] = 0 \quad \text{for all } i, j \\ \text{Var}[d_{ij}] &= \mathbb{E}[(d_{ij} - \mathbb{E}[d_{ij}])^2] = \mathbb{E}[d_{ij}^2] = \sigma^2 \quad \text{for all } i, j \\ \text{Cov}[d_{ij}, d_{km}] &= \mathbb{E}[(d_{ij} - \mathbb{E}[d_{ij}])(d_{km} - \mathbb{E}[d_{km}])] = \begin{cases} \sigma^2 & : \quad i, j = k, m \\ 0 & : \quad \text{else} \end{cases} \end{aligned}$$

Here  $E[\cdot]$  denotes expectation, and  $\text{Var}[\cdot]$  and  $\text{Cov}[\cdot]$  are the variances and covariances of their arguments. The matrix elements complying to these conditions are called *independent identically distributed* (i.i.d.) and zero mean, and the matrix itself is called a zero mean i.i.d. random matrix.

Under these conditions, a *least squares estimate*<sup>3</sup> of matrix  $\mathbf{A}_0$  is performed by determining the error matrix  $\mathbf{D}$  that is lowest in Frobenius norm  $\|\mathbf{D}\|_F = \left(\sum_{ij} d_{ij}^2\right)^{1/2} \rightarrow \min$  and lowers the rank of matrix  $\mathbf{A}$  such that  $\text{rank}(\mathbf{A} - \mathbf{D}) = 8$  holds, i.e. the Frobenius norm serves as a metric.

If additionally the random matrix process  $\{\mathbf{D}\}$  is Gaussian (and this is *not* required for the least squares process to be reasonable!), this very solution is also the *maximum likelihood* (ML) estimate of  $\mathbf{A}_0$  and consequently of  $\mathbf{f}$  as well, with all the advantageous characteristics of ML estimation applying.

The answer to the question how this least squares estimate of  $\mathbf{A}_0$  is obtained is provided by the *Eckart-Young-Mirsky theorem* (Eckart & Young 1936, Mirsky 1960):

**Theorem 1.** Let  $\mathbf{A}$  be a  $N \times M$  matrix of rank  $\text{rank}(\mathbf{A}) = r$  and let  $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \sum_{i=1}^r s_i \mathbf{u}_i \mathbf{v}_i^T$  with singular values  $s_1 \geq s_2 \geq \dots \geq s_{r-1} > s_r > 0$  be the singular value decomposition of  $\mathbf{A}$ .

If  $k < r$  then  $\mathbf{A}_k = \sum_{i=1}^k s_i \mathbf{u}_i \mathbf{v}_i^T$  is the solution to the minimization problem

$$\|\mathbf{A} - \hat{\mathbf{A}}_0\|_F \rightarrow \min \quad \text{subject to} \quad \text{rank}(\hat{\mathbf{A}}_0) \stackrel{!}{=} k$$

In our case, the actual rank of  $\mathbf{A}$  is 9 and the desired rank is 8, so we have

$$\hat{\mathbf{A}} = \sum_{i=1}^8 s_i \mathbf{u}_i \mathbf{v}_i^T \quad \text{and} \quad \hat{\mathbf{D}} = s_9 \mathbf{u}_9 \mathbf{v}_9^T \quad (4)$$

Since we are ultimately looking for the vector  $\mathbf{f}$  that solves  $\hat{\mathbf{A}}\mathbf{f} = \mathbf{0}$ , we see from equation (4) that  $\mathbf{v}_9$  (the right singular vector corresponding to the smallest singular value  $s_9$ ) is the solution to our problem, since it is the only vector that is orthogonal to all of the vectors  $\mathbf{v}_i, i = 1, \dots, 8$ . This vector is, as it is known from the theory of singular value decomposition [7], identical to the eigenvector of  $\mathbf{A}^T \mathbf{A}$  corresponding to the smallest eigenvalue. Here, it becomes obvious that the solution of step 1 and step 2 both reduce to the rank reduction of a given matrix, controlled by a given (or tacitly assumed) error metric. Note that this proceeding, stacking two approximation problems on top of each other does not necessarily provide a rank 2 matrix that is optimum with respect to the metric defined in step 1.

Thus, we have apparently derived the very same result as it is known for a long time by making a considerable detour. However, we will show in the following that this detour is really worthwhile, since it opens the door to a very valuable theory, from which we can use methods and tools for adjusting our solution to

<sup>3</sup> Note that this is not a *linear* least squares problem.

the precise requirements given by the algebraic and statistical structure of the problem at hand. This framework is the theory of *total least squares estimation*, also known as *errors-in-variables-modeling* (EIV) or *orthogonal regression*.

## 4 Total Least Squares

What we have done so far is in fact to use the basic procedure of Total Least Squares estimation. [8–11]:

**Definition 1 (Total Least Squares problem).** *We are given an overdetermined set of  $N$  equations  $\mathbf{a}^T \mathbf{x} = b$  in  $M$  unknowns  $\mathbf{x}$ , compiled to a matrix equation  $\mathbf{A} \mathbf{x} = \mathbf{b}$ . Both the vector  $\mathbf{b}$  as well as the matrix  $\mathbf{A}$  are subject to errors. The total least squares problem consists in finding an  $N \times M$  matrix  $\hat{\mathbf{A}}$  and an  $M$ -vector  $\hat{\mathbf{b}}$  for which the equation*

$$\hat{\mathbf{A}} \mathbf{x} = \hat{\mathbf{b}} \quad (5)$$

*has an exact solution, (i.e.  $\hat{\mathbf{b}}$  is in the column space of  $\hat{\mathbf{A}}$ ) under the condition that the deviation between  $(\mathbf{A} | \mathbf{b})$  and  $(\hat{\mathbf{A}} | \hat{\mathbf{b}})$  is minimal in terms of the Frobenius norm:*

$$\|(\mathbf{A} | \mathbf{b}) - (\hat{\mathbf{A}} | \hat{\mathbf{b}})\|_F \rightarrow \min \quad (6)$$

*Once a minimizing approximation  $(\hat{\mathbf{A}} | \hat{\mathbf{b}})$  has been found, any vector  $\mathbf{x}$  satisfying eqn.(5) is called a TLS solution.*

In the last decade, this basic TLS problem has been extended to a considerable number of generalizations and specializations, such as *Generalized Total Least Squares (GTLS)*, *Structured Total Least Squares (STLS)* and many more, cf. [8–10, 12, 13]. One of the most important motivations for the development of these specialized versions of TLS is the need for a metric that differs from the Frobenius norm, either due to a prescribed structure of the true matrix  $\mathbf{A}_0$  (e.g. Toeplitz) or due to disturbances in  $\mathbf{A}$  that do not have the simple statistical structure given by a zero-mean i.i.d. matrix  $\mathbf{D}$ .

Van Huffel [10] gives a comprehensive overview on the statistical properties of TLS solutions, including conditions for consistency of the estimate, i.e. the requirement that the estimate converges towards the true parameter vector as the number of measurements is increased.

If the errors in  $\mathbf{A}$  and  $\mathbf{b}$  are uncorrelated with zero mean and equal variance, then under mild conditions the TLS solution  $\hat{\mathbf{x}}$  is a strongly consistent estimate of the solution of the unperturbed equation  $\mathbf{A}_0 \mathbf{x} = \mathbf{b}_0$ .

If, however, this is *not* the case, for instance if the errors are correlated and of unequal size, or if some columns of  $\mathbf{A}$  are error-free, an adequate estimate can be obtained by use of the Generalized Total Least Squares technique [10] which essentially consists in replacing the metric given in eqn.(6) by

$$\left\| \mathbf{W}_1 \cdot \left( (\mathbf{A} | \mathbf{b}) - (\tilde{\mathbf{A}} | \tilde{\mathbf{b}}) \right) \cdot \mathbf{W}_2 \right\|_F \rightarrow \min. \quad (7)$$

with suitably chosen weight matrices  $\mathbf{W}_1$  and  $\mathbf{W}_2$  which perform a so-called *equilibration*. The case of matrices  $\mathbf{A}$  known to contain *exact*, i.e. error-free columns is handled by Demmel [14].

We are now equipped with a sufficient repertoire of techniques to handle non-standard problems of type TLS; but let us first have a look on the subject of data normalization before we investigate the statistical structure of the specific problem we are dealing with.

## 5 Input data normalization

### 5.1 Normalized eight-point algorithm

The 'classic' 8+n-point algorithm is known to be very susceptible to noise. In 1995 Hartley pointed out that a considerable improvement in performance could be achieved by a normalization of the input data (point correspondences) and it was demonstrated by examples that the quality of the results of the normalized 8+n-point algorithm is significantly better than the standard procedure, and comparable to the best iterative algorithms [1]. A second paper [2] exposed the principle in more detail, describing several variants of data normalization. However, an analysis of the statistical input-output relation, referring to the central entities of any estimation procedure, namely bias, variance and consistence, is missing in both of these papers.

Hartley's proposal is centred in the observation that a valid solution for the initially given problem  $\mathbf{A}\mathbf{f} = \mathbf{0}$  (which is, as we know, a TLS problem) can be obtained as well, if the initially given correspondence data which may be specified in arbitrary coordinate frames are linearly transformed to a canonical coordinate frame which is derived from the distribution of the data. Let  $\mathbf{R}$  and  $\mathbf{S}$  be two non-singular  $3 \times 3$  matrices which are used to obtain the transformed correspondence data  $\mathbf{v}'_i = \mathbf{S}\mathbf{v}_i$  and  $\mathbf{u}'_i = \mathbf{R}\mathbf{u}_i$  from the initially given data  $\mathbf{u}_i$  and  $\mathbf{v}_i$ . Instead of looking for a solution to  $\mathbf{v}_i^T \mathbf{F} \mathbf{u}_i = 0$ , a solution for

$$\underbrace{\mathbf{v}'_i^T}_{\mathbf{v}'_i} \underbrace{\mathbf{S}^T (\mathbf{S}^T)^{-1} \mathbf{F} \mathbf{R}^{-1}}_{\mathbf{F}'} \underbrace{\mathbf{R} \mathbf{u}_i}_{\mathbf{u}'_i} = 0. \quad (8)$$

is sought. In the new coordinate frame, matrix  $\mathbf{F}'$  is determined the same way as before: Find a TLS solution, rearrange it in matrix form and enforce the rank 2 constraint using the Eckart-Young-Mirsky-theorem. The inverse transformation  $\mathbf{F} = \mathbf{S}^T \mathbf{F}' \mathbf{R}$  will yield the desired matrix  $\mathbf{F}$ . Hartley points out that the TLS solution of eqn.(1) is not identical to the TLS solution of eqn.(8), and this is really what is intended: By selection of proper matrices  $\mathbf{R}$  and  $\mathbf{S}$  the initial TLS can be solved under a problem-adapted metric.<sup>4</sup>

<sup>4</sup> Unfortunately, the coordinate transform can help only in step 1 of the estimation process, but not in step 2 (reduction to rank 2).

## 5.2 Choice of the transformation matrices

Hartley proposes to apply the transformations both to vectors  $\{\mathbf{u}_i\}$  and vectors  $\{\mathbf{v}_i\}$ . We will now describe the transformation  $\mathbf{v}'_i = \mathbf{S}\mathbf{v}_i$ . Since the third component of the homogeneous vector should not be changed, possible transformations will consist of a translation, a rotation and a scaling in the first two components.

$$\mathbf{S} = \text{diag}\{\alpha_1, \alpha_2, 1\} \cdot \begin{pmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -\frac{1}{N} \sum_{i=1}^N v_{i1} \\ 0 & 1 & -\frac{1}{N} \sum_{i=1}^N v_{i2} \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

The righthand matrix translates the new coordinate frame into the center of gravity of the point set. The middle matrix may rotate the coordinate frame and the lefthand matrix is a scaling matrix.

Hartley proposes two different normalizing schemes. Whereas the translation and the rotation into the principal axes frames are identical in his papers, the scaling is not. In [1] the scaling is designed with the aim to make the mean *absolute value* of the new coordinates equal to 1, whereas in [2] the points are normalized with the aim to make the mean *square* of the coordinates equal to 1. In our experiments we have found only very small differences between the results obtained with these two normalization schemes<sup>5</sup>, which we will denote by **Hartley1** and **Hartley2**.

## 6 Perturbation of the TLS solution

As the determination of the SVD, which is the central operation in our algorithm, involves the solution of an eigensystem problem, we will now consider the influence of an error matrix on the eigenvectors of the unperturbed matrix.

### 6.1 Perturbation of the eigenvectors

Golub & van Loan [9, chapter 7.2.4] give a linear approximation of the perturbation in eigenvectors. But whereas they regard arbitrary  $N \times N$ -matrices with complex elements we are only interested in symmetric matrices with real elements. Therefore their equations simplify considerably.

**Theorem 2.** *Let  $\mathbf{B} \in \mathbb{R}^{n \times n}$  be a symmetric matrix with distinct eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  and eigenvectors  $\mathbf{x}_k$ . Assume  $\mathbf{C} \in \mathbb{R}^{n \times n}$  satisfies  $\|\mathbf{C}\|_2 = 1$ . We are looking for the eigenvectors of  $\mathbf{B} + \varepsilon\mathbf{C}$ .*

*A Taylor expansion of  $\mathbf{x}_k(\varepsilon)$  has the following form:*

$$\mathbf{x}_k(\varepsilon) = \mathbf{x}_k + \delta_{\mathbf{x}_k} + O(\varepsilon^2) \quad (10)$$

---

<sup>5</sup> Hartley also proposes an algorithm with isotropic scaling and no rotation at all in his second paper, which has not been considered here.

with

$$\delta_{\mathbf{x}_k} = \sum_{\substack{i=1 \\ i \neq k}}^n \frac{\mathbf{x}_i^T \mathbf{C} \mathbf{x}_k}{(\lambda_k - \lambda_i)} \mathbf{x}_i = \left( \sum_{\substack{i=1 \\ i \neq k}}^n \frac{\mathbf{x}_i \mathbf{x}_i^T}{(\lambda_k - \lambda_i)} \right) \mathbf{C} \mathbf{x}_k \quad (11)$$

Thus, we have made a very important observation: If  $E[\mathbf{C}] = \mathbf{0}$  holds, it follows that  $E[\delta_{\mathbf{x}_k}] = \mathbf{0}$ :

$$E[\delta_{\mathbf{x}_k}] = E \left[ \left( \sum_{\substack{i=1 \\ i \neq k}}^n \frac{\mathbf{x}_i \mathbf{x}_i^T}{(\lambda_k - \lambda_i)} \right) \mathbf{C} \mathbf{x}_k \right] = \left( \sum_{\substack{i=1 \\ i \neq k}}^n \frac{\mathbf{x}_i \mathbf{x}_i^T}{(\lambda_k - \lambda_i)} \right) E[\mathbf{C}] \mathbf{x}_k = \mathbf{0} . \quad (12)$$

Now we have shown that

$$E[\text{EigVec}(\mathbf{B} + \varepsilon \mathbf{C})] = \text{EigVec}(\mathbf{B}) \quad \text{if} \quad E[\mathbf{C}] = \mathbf{0} , \quad (13)$$

where  $\text{EigVec}(\cdot)$  denotes the set of eigenvectors of a given matrix.

## 6.2 Perturbation of the singular vectors

We assume that the vectors  $\mathbf{u}_i$  in the first image are error-free whereas the corresponding point coordinates  $\mathbf{v}_i$  in the second image are corrupted by zero-mean measurement errors, i.e.  $\mathbf{u}_i = \mathbf{u}_{i0}$  and  $\mathbf{v}_i = \mathbf{v}_{i0} + \mathbf{b}_i$  with  $E[\mathbf{b}_i] = \mathbf{0}$ . Since  $\mathbf{b}_i$  is zero-mean, we have  $\text{Cov}[\mathbf{b}_i] = \mathbf{C}_{b_i} = E[\mathbf{b}_i \mathbf{b}_i^T]$ , and in the absence of further information we may assume  $\mathbf{C}_{b_i} = \text{diag}\{\sigma_b^2, \sigma_b^2, 0\}$ . The row vectors of  $\mathbf{A}$  (called  $\mathbf{a}_i^T$ ) are corrupted with an error  $\mathbf{d}_i^T$  ( $\mathbf{a}_i = \mathbf{a}_{i0} + \mathbf{d}_i$ ) and for  $\mathbf{d}_i$  we get:

$$\mathbf{d}_i^T = (b_{i1}u_{i01}, b_{i1}u_{i02}, b_{i1}, b_{i2}u_{i01}, b_{i2}u_{i02}, b_{i2}, 0, 0, 0) . \quad (14)$$

We see that the last three elements are error-free. It is seen easily that  $E[\mathbf{d}_i] = \mathbf{0}$  for all  $i$  holds as well.

We are looking for the eigenvectors<sup>6</sup> of  $\mathbf{A}_0^T \mathbf{A}_0$  which we denote as  $\mathbf{x}_{i0}$  (actually we are only interested in the eigenvector  $\mathbf{x}_{90}$  corresponding to the smallest eigenvalue).

$$\mathbf{A}^T \mathbf{A} = \mathbf{A}_0^T \mathbf{A}_0 + \underbrace{\mathbf{A}_0^T \mathbf{D} + \mathbf{D}^T \mathbf{A}_0}_{\Delta_1} + \underbrace{\mathbf{D}^T \mathbf{D}}_{\Delta_2} = \mathbf{A}_0^T \mathbf{A}_0 + \Delta \quad (15)$$

$\mathbf{A}_0^T \mathbf{A}_0$  is perturbed with two different error matrices; a term  $\Delta_1$  which is linear in  $\mathbf{D}$  (with  $E[\Delta_1] = \mathbf{0}$ ) and a quadratic term  $\Delta_2$  (with  $E[\Delta_2] \neq \mathbf{0}$ ). Since  $\mathbf{A}_0^T \mathbf{A}_0$  is symmetric, we can apply eqn.(11) to  $\delta_{\mathbf{x}_0}$  and get (for small  $\mathbf{D}$ ) an expression for the error in  $\mathbf{x}_0$

$$\delta_{\mathbf{x}_0} = \left( \sum_{i=1}^8 \frac{\mathbf{x}_{i0} \mathbf{x}_{i0}^T}{(\lambda_{90} - \lambda_{i0})} \right) \Delta \cdot \mathbf{x}_{90} \quad (16)$$

In order to obtain an unbiased estimator we have to ensure that  $E[\delta_{\mathbf{x}_0}] \stackrel{!}{=} \mathbf{0}$  holds. In general this will *not* be true due to  $E[\Delta] \neq \mathbf{0}$ .

<sup>6</sup> The subscript 0 denotes the *true* eigenvalues and eigenvectors of  $\mathbf{A}_0^T \mathbf{A}_0$ .



### 6.3 Requirements for the statistical structure of matrix $\mathbf{A}$

We can formulate the last requirement in another way as well: we would like to achieve that the expected eigenvectors (at least the ninth) of the perturbed matrix do not differ from the eigenvectors of the unperturbed matrix. Thus the stochastic process that generates the perturbation matrices  $\Delta$  should be such that

$$\mathbb{E} [\text{EigVec} (\mathbf{A}_0^T \mathbf{A}_0 + \Delta)] \stackrel{!}{=} \text{EigVec} (\mathbf{A}_0^T \mathbf{A}_0) \quad (17)$$

holds. In section 6 we have already derived with eqn. (13) that eqn. (17) would hold in linear approximation if  $\mathbb{E} [\Delta] = \mathbf{0}$ , which is not the case. However, we will see that even a weaker requirement is sufficient in order to make eqn. (17) hold.

**Proposition:** If  $\mathbb{E} [\Delta] = c\mathbf{I}$  holds, the requirement (17) assuring unbiasedness holds as well.

In order to prove this proposition, we first note that  $\text{EigVec} (\mathbf{A} + c\mathbf{I}) = \text{EigVec} (\mathbf{A})$  for any matrix  $\mathbf{A}$  ([15]). Now we define  $\Delta' = \Delta - c\mathbf{I}$ . Obviously  $\mathbb{E} [\Delta'] = \mathbf{0}$  holds, thus eqn.(13) can be applied and we obtain:

$$\begin{aligned} \mathbb{E} [\text{EigVec} (\mathbf{A}_0^T \mathbf{A}_0 + \Delta)] &= \mathbb{E} [\text{EigVec} ((\mathbf{A}_0^T \mathbf{A}_0 + c\mathbf{I}) + \Delta')] \\ &= \text{EigVec} (\mathbf{A}_0^T \mathbf{A}_0 + c\mathbf{I}) = \text{EigVec} (\mathbf{A}_0^T \mathbf{A}_0) . \end{aligned}$$

Therefore, we have to ensure that

$$\mathbb{E} [\Delta] = \mathbb{E} [\Delta_1] + \mathbb{E} [\Delta_2] = \mathbb{E} [\mathbf{A}_0^T \mathbf{D} + \mathbf{D}^T \mathbf{A}_0] + \mathbb{E} [\mathbf{D}^T \mathbf{D}] \stackrel{!}{=} c\mathbf{I} \quad (18)$$

holds. From  $\mathbb{E} [\mathbf{D}] = \mathbf{0}$  follows  $\mathbb{E} [\Delta_1] = \mathbf{0}$ , and therefore our requirement can be transformed into the simple form

$$\mathbb{E} [\mathbf{D}^T \mathbf{D}] \stackrel{!}{=} c\mathbf{I} \quad (19)$$

We see that, for small errors, it does not depend on the *linear* perturbation terms whether the estimator is unbiased but solely on the *quadratic* term  $\mathbf{D}^T \mathbf{D}$ .

Let us now regard the matrix  $\mathbf{D}^T \mathbf{D}$ . Its element  $(j, k)$  is obtained from the scalar product of the  $j$ -th and the  $k$ -th column of  $\mathbf{D}$ :  $(\mathbf{D}^T \mathbf{D})_{jk} = \sum_{i=1}^N d_{ij} d_{ik}$ . Therefore our requirement is  $\sum_{i=1}^N \mathbb{E} [d_{ij} d_{ik}] \stackrel{!}{=} \delta_{jk} c$ . In matrix form we can express eqn. (19) as follows:

$$\sum_{i=1}^N \text{Cov} [\mathbf{a}_i] = \sum_{i=1}^N \mathbb{E} [\mathbf{d}_i \mathbf{d}_i^T] = \sum_{i=1}^N \mathbf{C}_{d_i} \stackrel{!}{=} c\mathbf{I} . \quad (20)$$

If we look at  $\mathbf{d}_i$  (eqn.(14)) we realize that the last three columns of  $\mathbf{A}$  are error-free and therefore in  $\mathbf{C}_{d_i} = \text{Cov} [\mathbf{a}_i]$  the last three columns and rows will be 0.

However, this is no serious problem. On the contrary, we can turn it into an advantage since we have *more* information about the possible error structure in

matrix  $\mathbf{A}$ . We can replace the normal TLS estimation procedure with the already mentioned special form of TLS that minimizes the error matrix norm subject to the constraint that a certain number of columns (here: three) are free of errors. Two different but equivalent TLS-FC (Total Least Squares - Fixed Columns) algorithms are described by Golub, Hoffman & Stewart [16] and Demmel [14]. We use the second algorithm which is based on the SVD of the matrix  $\mathbf{A}$ .

Our modified requirement is:

$$\sum_{i=1}^N \mathbf{C}_{d_i} \stackrel{!}{=} c\tilde{\mathbf{I}} \quad \text{with} \quad \tilde{\mathbf{I}} = \text{diag}\{1, 1, 1, 1, 1, 1, 0, 0, 0\} \quad (21)$$

## 7 Derivation of optimum normalization transforms

Now we recall Hartley's idea and perform a transformation of the input data  $\mathbf{u}_i$  and  $\mathbf{v}_i$ :

$$\mathbf{u}'_i = \mathbf{R}\mathbf{u}_i = \mathbf{R}\mathbf{u}_{i0} = \mathbf{u}'_{i0} \quad \text{and} \quad \mathbf{v}'_i = \mathbf{S}\mathbf{v}_i = \mathbf{S}\mathbf{v}_{i0} + \mathbf{S}\mathbf{b}_i = \mathbf{v}'_{i0} + \mathbf{b}'_i$$

In contrast to Hartley we will now deduce the matrices  $\mathbf{R}$  and  $\mathbf{S}$  from the requirement we set up in the last section. The matrices  $\mathbf{R}$  and  $\mathbf{S}$  must be non-singular (otherwise eqn.(8) could not hold) and have the following structure

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad (22)$$

The third rows are (0,0,1) because the third component of the homogenous vectors  $\mathbf{u}_i$  and  $\mathbf{v}_i$  shall remain 1. Now we look at the expectation and the covariance matrix of  $\mathbf{b}'_i$ :

$$\mathbf{E}[\mathbf{b}'_i] = \mathbf{E}[\mathbf{S}\mathbf{b}_i] = \mathbf{S}\mathbf{E}[\mathbf{b}_i] = \mathbf{S}\mathbf{0} = \mathbf{0} \quad (23)$$

$$\begin{aligned} \mathbf{C}_{b'} &= \text{Cov}[\mathbf{b}'_i] = \mathbf{E}[\mathbf{b}'_i\mathbf{b}'_i{}^T] = \mathbf{E}[\mathbf{S}\mathbf{b}_i\mathbf{b}_i{}^T\mathbf{S}^T] = \mathbf{S}\mathbf{E}[\mathbf{b}_i\mathbf{b}_i{}^T]\mathbf{S}^T \\ &= \mathbf{S} \begin{pmatrix} \sigma_b^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{S}^T = \begin{pmatrix} c_{11} & c_{12} & 0 \\ c_{21} & c_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (24)$$

After considering the statistical properties of the transformed input data we can now use this knowledge. In the new coordinate frame we have

$$\mathbf{d}'_i{}^T = (b'_{i1}u'_{i1}, b'_{i1}u'_{i2}, b'_{i1}, b'_{i2}u'_{i1}, b'_{i2}u'_{i2}, b'_{i2}, 0, 0, 0). \quad (25)$$

The vector  $\mathbf{d}'_i$  is zero-mean because  $\mathbf{b}'_i$  is zero-mean (see eqn.(23)). We obtain:

$$\text{Cov}[\mathbf{d}'_i] = \mathbf{C}_{d'_i} = \begin{pmatrix} c_{11}\mathbf{X}_i & c_{12}\mathbf{X}_i & 0 \\ c_{21}\mathbf{X}_i & c_{22}\mathbf{X}_i & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{with} \quad \mathbf{X}_i \stackrel{def}{=} \begin{pmatrix} u'_{i1}{}^2 & u'_{i1}u'_{i2} & u'_{i1} \\ u'_{i1}u'_{i2} & u'_{i2}{}^2 & u'_{i2} \\ u'_{i1} & u'_{i2} & 1 \end{pmatrix} \quad (26)$$

Using the Kronecker matrix product  $\otimes$  this can be written as:

$$\mathbf{C}_{d'_i} = \mathbf{C}_{b'} \otimes \mathbf{X}_i \quad (27)$$

With the requirement expressed by eqn.(21) we obtain:

$$\sum_{i=1}^N \mathbf{C}_{d'_i} = \sum_{i=1}^N \mathbf{C}_{b'} \otimes \mathbf{X}_i = \mathbf{C}_{b'} \otimes \left( \sum_{i=1}^N \mathbf{X}_i \right) \stackrel{!}{=} c \tilde{\mathbf{I}} \quad (28)$$

This means that the following two equations must hold simultaneously:

$$\mathbf{C}_{b'} = \mathbf{S} \cdot \mathbf{C}_d \cdot \mathbf{S}^T \stackrel{!}{=} \begin{pmatrix} k_1 & 0 & 0 \\ 0 & k_1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (29)$$

$$\sum_i \mathbf{X}_i = \begin{pmatrix} \sum_i u'_{i1} u'_{i1} & \sum_i u'_{i1} u'_{i2} & \sum_i u'_{i1} \\ \sum_i u'_{i2} u'_{i1} & \sum_i u'_{i2} u'_{i2} & \sum_i u'_{i2} \\ \sum_i u'_{i1} & \sum_i u'_{i2} & \sum_i 1 \end{pmatrix} = \sum_i \mathbf{R} \mathbf{u}_i \mathbf{u}_i^T \mathbf{R}^T \stackrel{!}{=} \begin{pmatrix} k_2 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_2 \end{pmatrix} \quad (30)$$

This can also be seen from eqn.(26). Since  $\mathbf{C}_b$  is already assumed to be  $\text{diag} \{ \sigma_b^2, \sigma_b^2, 0 \}$ ,  $\mathbf{S}$  may be an arbitrary matrix of the following structure:

$$\mathbf{S} = \alpha \cdot \begin{pmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix}$$

with arbitrary values for the parameters  $\alpha, \phi, t_1, t_2$  except  $\alpha \neq 0$ . This looks very similar to the transformation given by eqn.(9) except for the fact that  $\alpha$  is no diagonal matrix, i.e. only *isotropic* scaling is allowed for the vectors  $\mathbf{v}_i$ . In contrast to this requirement Hartley is performing an *anisotropic* scaling of the 'second' vector  $\mathbf{v}_i$  as well. However, this will cause problems only if two principal moments of the vectors  $\mathbf{v}_i$  are notably different.

We continue regarding eqn.(30). If we require that the third component of vector  $\mathbf{u}'_i$  remains to be 1, this fixes the otherwise arbitrary constant  $k_2$  to be  $N$ . Solutions for  $\mathbf{R}$  can be found by considering that from eqn.(30) and the requirement that  $\mathbf{R}$  be nonsingular we obtain

$$\sum_i \mathbf{u}'_i \mathbf{u}'_i{}^T = N \cdot \mathbf{R}^{-1} (\mathbf{R}^T)^{-1} \quad (31)$$

This fixes the  $3 \times 3$ -matrix  $\mathbf{R}$  except for a premultiplication by an arbitrary orthonormal  $3 \times 3$  matrix. In any case, from eqn.(30) we clearly see that the matrix  $\mathbf{R}$  must inevitably yield

$$\sum_i \mathbf{u}'_{i1} = \sum_i \mathbf{u}'_{i2} = 0 \quad (32)$$

$$\sum_i (\mathbf{u}'_{i1})^2 = \sum_i (\mathbf{u}'_{i2})^2 = N \quad (33)$$

$$\sum_i \mathbf{u}'_{i1} \mathbf{u}'_{i2} = 0 \quad (34)$$

These are exactly the non-isotropic scaling prescriptions proposed by Hartley [2] as one of several options. Our derivation has shown that this scaling process is absolutely necessary *for the vectors  $\mathbf{u}_i$* . Finally, the difference between the method derived in our paper and Hartley’s method is that firstly the transformation  $\mathbf{S}$  is isotropic, and secondly that we use the TLS-FC algorithm instead of plain TLS.

## 8 Experimental results

The different approaches discussed so far have been implemented in *Mathematica*, using synthetic data with an exactly known setup and exactly controllable measurement errors. We would like to stress the point that these experimental results are reported here in order to illustrate several effects that have been observed in a very large set of experiments. Due to the huge number of parameters, the derivation of generally valid conclusions is only possible on the basis of theoretical considerations, and experiments are to be understood as attempts to check whether the theory can be in accordance with reality.

We considered the case of the essential matrix (known camera parameters, enabling us to analyze the algorithms’ performance in a Euclidean space) and performed a large number of simulations for different settings of the true motion (by now: pure translation in varying directions) and using ‘objects’ determined by a set of  $N$  random points inside of a cube given by world coordinates  $x, y, z \in [-1, 1]$ . These points have been projected onto the image plane yielding a set of  $N$  vector pairs  $\mathbf{u}, \mathbf{v}$ . Examples of such point sets visualized as displacement vector fields are shown in fig. 1 and fig. 8.

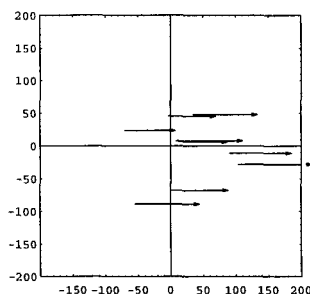
Errors in the point correspondences have been introduced by adding noise vectors  $(d_1, d_2)$  drawn from a uniform distribution  $d_1, d_2 \in [-\epsilon/2, \epsilon/2]$  to the  $\mathbf{v}_i$  vectors ( $\mathbf{u}_i$  remaining undisturbed) with  $\epsilon$  varying in the range between 0 and 30 units. Thus the variance of the error components  $d_i$  is  $\epsilon^2/12$ . The focal length was 256 units. The virtual image area had a width and height of 400 units.

For a fixed motion situation and a fixed ‘object’, 500 experiments have been performed for each of 10 different values of  $\epsilon$ , respectively. What we were most interested in was the difference between the true motion vector direction and the estimates of this entity as produced by the four different algorithm variations which have been under investigation. As the translation direction can only be computed up to a scale factor, we used the angular difference between the two unit vectors (true and estimated direction) as a measurement of quality.

We did not only compute the mean angular error (MAE) between the estimated and the true direction of translation (each simulation yields one angular error, MAE = mean of all these errors), but also the angular error between the true translation direction and the mean estimated translation direction (compute mean estimated direction *first* and *then* the angular error of this quantity; AEM = angular error of mean direction). Whereas the MAE reflects the overall dispersion of the estimation result, the AEM measure will flag situations where the expectation of the estimated direction is not in coincidence with the true di-

rection of translation. Thereby a more differentiated analysis of the algorithms' output is possible and cases where the considered algorithm is biased can be detected. The four algorithm variants considered here are

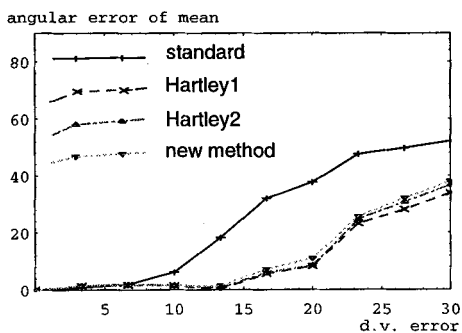
- standard algorithm: no normalization; application of plain TLS for first estimation step and subsequent rank reduction.
- “Hartley1”: anisotropic scaling of both  $\mathbf{u}$  and  $\mathbf{v}$ ; application of plain TLS for first estimation step and subsequent rank reduction.
- “Hartley2”: anisotropic normalization of both  $\mathbf{u}$  and  $\mathbf{v}$ ; application of plain TLS for first estimation step and subsequent rank reduction.
- new algorithm: anisotropic normalization of  $\mathbf{u}$ , shift to the center of mass and isotropic scaling for  $\mathbf{v}$ ; application of TLS with fixed columns according to [14].



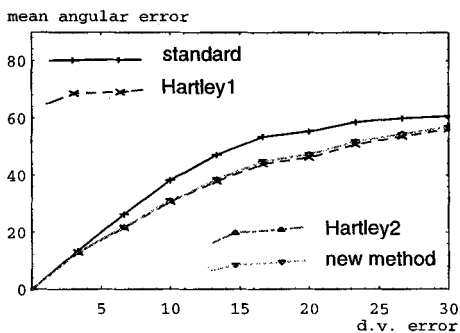
**Fig. 1.** Example of a displacement vector field used in our experiments; motion parallel to image plane ( $x$  direction),  $N=9$

In our experiments, the theoretically predicted problems with the standard algorithm and the two variants proposed by Hartley did in fact occur. We list the different observations:

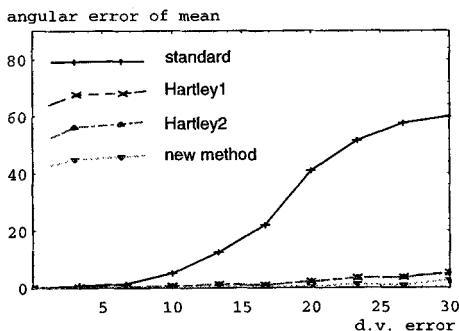
- The standard algorithm (solid line in all figures) is extremely susceptible to noise in the input data. Even for a rather moderate amount of error in the correspondences, the mean angular error increases very rapidly and settles in the area of about 60 degrees. It can be shown rather easily that the mean angular error to be expected when the direction of translation is *guessed* (i. e. uniform distribution on the unit sphere), is about 57 degrees. So we can only join the choir of authors stating that the plain  $8+n$ -points algorithm is virtually useless due to its lack of stability.
- The standard algorithm is strongly biased towards the viewing direction. This is a known – but not widely known – fact (cf. [17] and the related case discussed in [18]). This bias gets visible from a comparison of figures 2, 4,



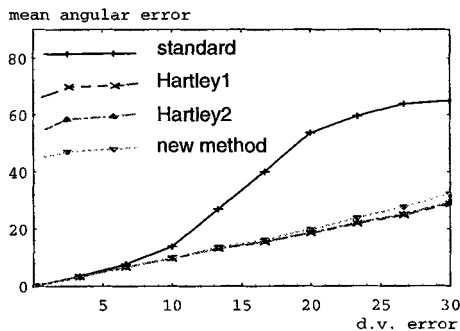
**Fig. 2.** Angular difference of mean direction estimate to true direction (AEM); here: motion in  $x$  direction,  $N=9$



**Fig. 3.** Mean angular difference between estimated and true motion direction (MAE); motion type =  $x$  direction,  $N=9$



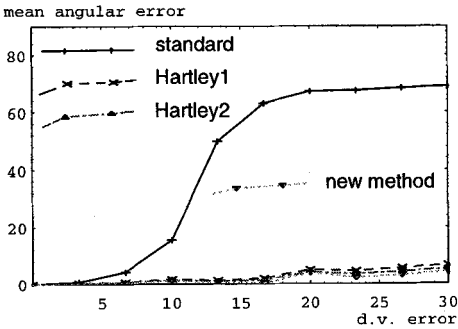
**Fig. 4.** AEM; here: motion in  $x$  direction,  $N=18$



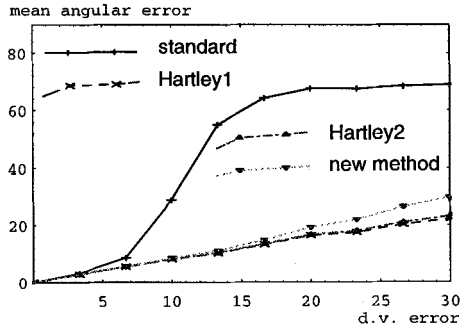
**Fig. 5.** MAE; motion type =  $x$  direction,  $N=18$

and 6 (translation parallel to image plane; 9, 18 and 36 correspondences) versus figure 9 (translation in viewing direction, 9 correspondences) which is representative for the behaviour of 18, 36 and 60 correspondences as well. The apparently excellent behaviour of the standard algorithm with respect to the AEM measure in fig. 9 is caused by the fact that the true motion direction is in exact coincidence with the bias of the estimator.

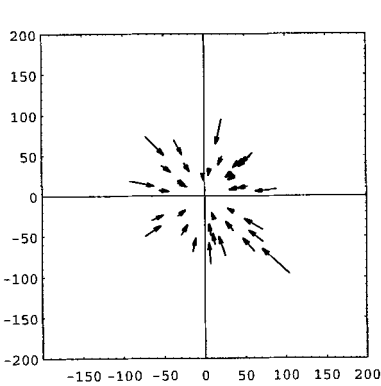
- The performance of Hartley's variants with respect to the AEM criterion does not provide any clue for a bias (cf. figs. 2, 4, 6, and 9), and neither does our new algorithm, which is almost indistinguishable from the latter two with respect to the AEM criterion. Thus the results of the experiments are compatible with our claim that the normalization procedure eliminates the bias, as indented.
- When we performed the first experiments, we were rather astonished to stumble upon the strange behaviour of the 'Hartley-normalized' algorithms with respect to the mean angular error (MAE) for the case that the direction of translation is identical to the viewing direction (translation in



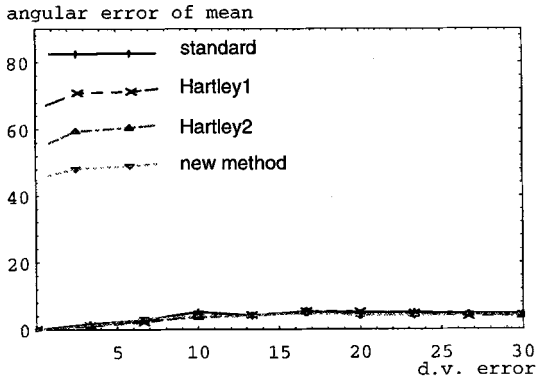
**Fig. 6.** AEM; here: motion in  $x$  direction,  $N=36$



**Fig. 7.** MAE; motion type =  $x$  direction,  $N=36$



**Fig. 8.** Displacement vector field for motion in  $z$  direction (in viewing direction),  $N=36$



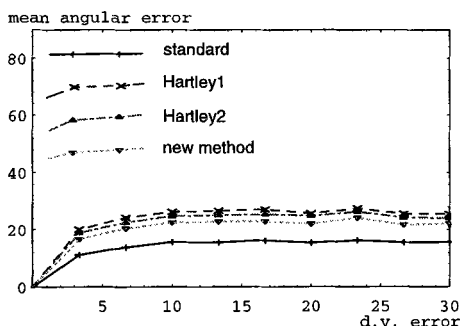
**Fig. 9.** AEM; here: motion in  $z$  direction,  $N=9$

$z$ -direction): Comparing figs.10, 11, 12, and 13 one finds that the MAE, and therefore the uncertainty of the estimation result, *increases* if more data (point correspondences) are added. Obviously, the regarded estimator is not consistent.

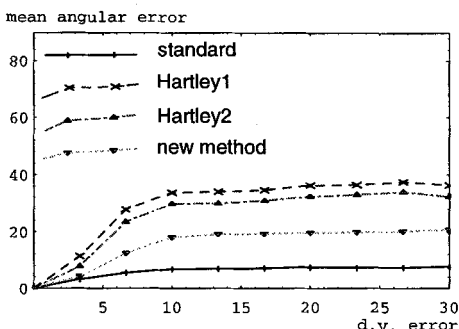
- For the method proposed in this paper, the MAE remains moderate for all situations investigated here, and - more important - it does not increase with increasing number of point correspondences as can be seen in figures 10 - 13.

## 9 Conclusions

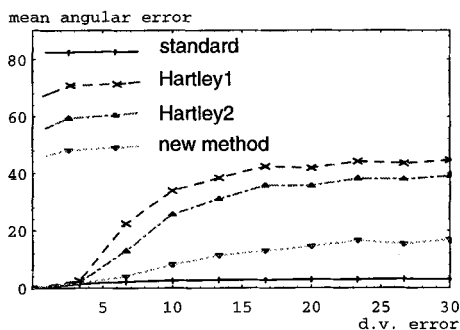
We have shown that a proper analysis of the statistical structure of errors in the input data of a motion estimation algorithm can provide valuable information about the sensitive parts of a given procedure. Moreover, this type of error



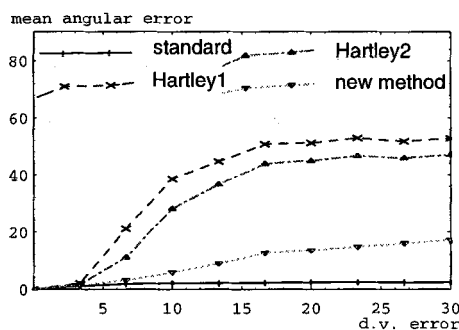
**Fig. 10.** MAE; motion type =  $z$  direction,  $N=9$



**Fig. 11.** MAE; motion type =  $z$  direction,  $N=18$



**Fig. 12.** MAE; motion type =  $z$  direction,  $N=36$



**Fig. 13.** MAE; motion type =  $z$  direction,  $N=60$

analysis can be the starting point for modifications that provide the algorithm with extensions that significantly improve its overall performance in terms of precision, stability and reliability.

We are convinced that the concept of Total Least Squares estimation will become established in the field of motion analysis, since many estimation tasks currently under investigation in various research groups do precisely belong to the class of problems the TLS method is aiming at.

It shall not be held back that even in the context of the specific problem discussed in this paper there are still several open questions, for instance the optimum estimation of a rank 2 fundamental matrix. For the moment being, we must confine ourselves to the remark that the Frobenius norm is certainly *not* the correct metric for performing the constraint enforcement step; instead, the correct metric is implicitly given by the covariance structure of the F-matrix estimate obtained in the first step. Here, further investigations are required and we hope to be able to present results before long.

Finally, we would like to express our confidence in that non-iterative algorithms for the two-view motion analysis problem will eventually perform at least



as well as the best iterative procedures as soon as a comprehensive exploitation of the statistical input-output relation has been performed. As it has been pointed out in Hartley's 1997 paper, the gap to be bridged is not very large.

## References

1. Hartley, R.I.: In defence of the 8-point-algorithm. In: Proc. Fifth International Conference on Computer Vision, IEEE Computer Society Press, S.1064-1070, June 1995.
2. Hartley, R.I.: In defence of the 8-point-algorithm. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, no.6, June 1997.
3. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. *Nature*, vol. 293, September 1981, S.433-435
4. Tsai, R.Y.; Huang, T.S.: Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. IEEE Trans. Pattern Analysis and Machine intelligence, vol.6, pp.13-27, 1984
5. Weng, J.; Huang, T.S.; Ahuja, N.: Motion and structure from image sequences. Springer Verlag, Series in Information Science, vol. 29, 1993.
6. Maybank, S.J.: The projective geometry of ambiguous surfaces. Phil. Trans. R.Soc.Lond., vol.A332, pp.1-47, 1990
7. Strang, G.: Linear algebra and its applications. 3rd edition, Harcourt Brace Jovanovich, Inc., 1988.
8. Golub, G.H.; van Loan, C.F.: An analysis of the total least squares problem, SIAM J. Matrix Anal. Appl., Vol. 17, No. 6, pp 883-893, 1980.
9. Golub, G.H.; van Loan, C.F.: Matrix computations. John Hopkins University Press 2nd ed., 1989.
10. van Huffel, S., Vandewalle, J.: The total least squares problem: Computational aspects and analysis. SIAM (Society for Industrial and Applied Mathematics), Philadelphia, 1991
11. Therrien, C.W.: Discrete random signals and statistical signal processing. Prentice Hall, 1992.
12. van Huffel, S.; Vandewalle, J.: Analysis and properties of the generalized total least squares problem  $AX \approx B$ , when some or all columns are subject to error. SIAM J. Matrix Anal. Appl., Vol. 10, no. 3, pp 294-315, 1989.
13. Mendel, J.: Lessons in estimation theory for signal processing, communications, and control. Prentice Hall, Englewood Cliffs, 1995
14. Demmel, J.W.: The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems. SIAM J. Numer. Anal. 24, pp. 199-206.
15. Rao, C.R.; Toutenburg, H.: Linear models: least squares and alternatives. Springer Series in Statistics, 1995.
16. Golub, G.H.; Hoffman, A.; Stewart, G.W.: A generalization of the Eckart-Young-Mirsky Matrix Approximation Theorem, Linear Algebra and its Applications, 88/89: 317-327, 1987
17. Kanatani, K.; Geometric computation for machine vision: a geometric viewpoint. Oxford University Press, U.K., 1993.
18. Earnshaw, A.M.; Blostein, S.D.: The performance of camera translation direction estimators from optical flow: analysis, comparison, and theoretical limits. IEEE Trans. Patt. Anal. and Machine Intell., vol.18, No. 9, 1996, pp.927-932