Estimation of Nonlinear Errors-in-Variables Models for Computer Vision Applications

Bogdan C. Matei, *Member*, *IEEE Computer Society*, and Peter Meer, *Senior Member*, *IEEE*

Abstract—In an errors-in-variables (EIV) model, all the measurements are corrupted by noise. The class of EIV models with constraints separable into the product of two nonlinear functions, one solely in the variables and one solely in the parameters, is general enough to represent most computer vision problems. We show that the estimation of such nonlinear EIV models can be reduced to iteratively estimating a linear model having point dependent, i.e., heteroscedastic, noise process. Particular cases of the proposed heteroscedastic errors-in-variables (HEIV) estimator are related to other techniques described in the vision literature: the Sampson method, renormalization, and the fundamental numerical scheme. In a wide variety of tasks, the HEIV estimator exhibits the same, or superior, performance as these techniques and has a weaker dependence on the quality of the initial solution than the Levenberg-Marquardt method, the standard approach toward estimating nonlinear models.

Index Terms—Nonlinear least squares, heteroscedastic regression, camera calibration, 3D rigid motion, uncalibrated vision.

1 Modeling Computer Vision Problems

Solving most computer vision problems requires the estimation of a set of parameters from noisy measurements using a *statistical model*. A statistical model provides a mathematical description of a problem in terms of a *constraint* equation relating the measurements to the parameters and of a *noise model* which characterizes the errors affecting the measurements [12, Section 1.1]. Under its most general form, the constraint can be multivariate, nonlinear, and with an implicit relationship between the ideal values of the measurements $\boldsymbol{z}_{io}$ and the ideal value of a parameter $\boldsymbol{\beta}_o$

$$\boldsymbol{f}(\boldsymbol{z}_{io}, \boldsymbol{\beta}_o) = \mathbf{0} \quad i = 1, \ldots, n \quad \boldsymbol{f}(\cdot) \in \mathbb{R}^m \quad \boldsymbol{z}_{io} \in \mathbb{R}^s \quad \boldsymbol{\beta}_o \in \mathbb{R}^q. \tag{1}$$

In computer vision problems, an additive measurement noise can usually be assumed. In the most general case of additive noise, the characteristics of the noise depend on the data point, i.e., the noise is *heteroscedastic*

$$\boldsymbol{z}_i = \boldsymbol{z}_{io} + \delta \boldsymbol{z}_i \quad \delta \boldsymbol{z}_i \sim GI(\mathbf{0}, \sigma_\nu^2 \mathrm{C}_{z_i}) \quad i = 1, \ldots, n \quad \boldsymbol{z}_{io} \in \mathbb{R}^s, \tag{2}$$

where $GI(\cdot)$ stands for a general, symmetric, and independent distribution whose first two central moments are available. The subscript "o" is used in the paper to distinguish the ideal, noise-free value of a quantity from its noisy measurement. The covariance matrices are assumed to be known only up to a common factor, the *noise variance* σ_ν^2.

A statistical model in which the noise affects *all* the elements of the data vector $\boldsymbol{z}_i$ is called an *errors-in-variables* (EIV) model. The EIV model includes the situation when some of the variables are not corrupted, i.e., their ideal value is available. Since these measurements are deterministic, the covariances $\sigma_\nu^2 \mathrm{C}_{z_i}$ become singular.

As will be shown later in the paper, in numerous 3D computer vision problems, the constraint (1) can be factorized into two parts. In general, these factors are nonlinear functions, the first depending only on the measurements and the second only on the parameter vector

$$\boldsymbol{f}(\boldsymbol{z}_{io}, \boldsymbol{\beta}_o) = \Phi(\boldsymbol{z}_{io})\boldsymbol{\theta}(\boldsymbol{\beta}_o) = \mathbf{0} \quad \Phi(\cdot) \in \mathbb{R}^{m \times p} \quad \boldsymbol{\theta}(\cdot) \in \mathbb{R}^p. \tag{3}$$

Such a constraint is called a *separable parameter*. Note that (3) can also be seen as linear in $\boldsymbol{\theta}$ if the dependence on $\boldsymbol{\beta}$ is not considered during the estimation process

$$\boldsymbol{f}(\boldsymbol{z}_{io}, \boldsymbol{\theta}_o) = \Phi(\boldsymbol{z}_{io})\boldsymbol{\theta}_o = \mathbf{0} \quad \boldsymbol{\theta}_o = \boldsymbol{\theta}(\boldsymbol{\beta}_o). \tag{4}$$

The estimation method proposed in the paper exploits this observation. Furthermore, many computer vision problems can already be solved without using an explicit reparameterization in $\boldsymbol{\beta}$.

It can be demonstrated that (4) can be generalized to account for the gauge matrix $\mathrm{H} \in \mathbb{R}^{p \times p}$ [21], [22], [31]

$$\Phi(\boldsymbol{z}_{io})\mathrm{H}\mathrm{H}^{-1}\boldsymbol{\theta}_o = \mathbf{0} \tag{5}$$

when the gauge matrix is orthonormal, $\mathrm{H}^\top \mathrm{H} = \mathrm{I}_p$. The constraint (4) also has an ambiguity to multiplication with a nonzero constant. To remove this ambiguity, an additional restriction has to be imposed on $\boldsymbol{\theta}_o$. Most often, $\boldsymbol{\theta}_o$ is assumed to have unit norm or one of its elements is set to a known value. The nature of the vision problem can impose additional constraints on either $\boldsymbol{\theta}_o$ or $\boldsymbol{\beta}_o$. We will discuss such problems in Section 5.

The ideal values of the measurements, $\boldsymbol{z}_{io}, i = 1, \ldots, n$, are, in general, unobservable and are called in the statistical literature *nuisance parameters* to distinguish them from $\boldsymbol{\beta}_o$, the main parameter of interest. The estimates of the nuisance

- *B.C. Matei is with the Vision Technologies Laboratory, Sarnoff Corporation, 201 Washington Road, Princeton, NJ 08540. E-mail: bmatei@sarnoff.com.*
- *P. Meer is with the Electrical and Computer Engineering Department, Rutgers University, 94 Brett Road, Piscataway, NJ 08854. E-mail: meer@caip.rutgers.edu.*

Manuscript received 30 Jan. 2006; accepted 6 Mar. 2006; published online 11 Aug. 2006.
Recommended for acceptance by D. Fleet.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0060-0106.

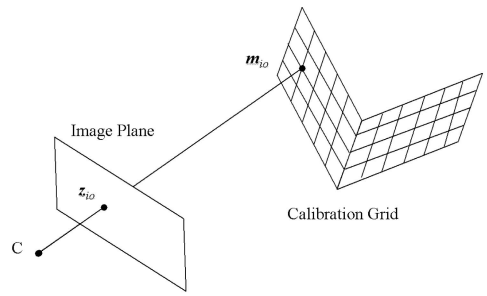


Fig. 1. Direct calibration of a camera. The coordinates of a 3D point $\boldsymbol{m}_{io}$ on a calibration grid are assumed known. Its projection $\boldsymbol{z}_i$ on the image plane is measured with error.

parameters, $\hat{\boldsymbol{z}}_i$, are the *corrected values* of the measurements and, often, finding them is the main goal of the computer vision task. Though the nuisance parameters can be treated as stochastic, such as is done in *structural* statistical models, in computer vision problems, it is difficult to make valid assumptions about their distribution, hence, we will treat them as fixed and unknown, as assumed in *functional* statistical models. The difference between structural and functional affects models more their respective statistical analysis, rather than the manner in which the estimates of parameters are obtained [36].

Numerous estimation techniques are available in the literature depending on the objective function selected, the manner in which the objective function is minimized, or on how the intrinsic constraints in the parameter are treated. Often, the objective functions are derived using the maximum-likelihood estimation theory; however, their actual minimization varies vastly due to simplifications made in order to make a problem tractable or the actual way in which the statistical properties of the noise are handled. Linear(ized) algorithms have the appeal of yielding close form solutions; however, they are notoriously sensitive in practice to errors in the measurements and result in biased solutions [19], [20]. Nonlinear objective functions have a better performance, however, they result in iterative algorithms which may not converge if initialized far away from the correct solution. A rigorous study on various choices made for the objective functions applied to solve 3D computer vision problems (structure from motion, triangulation) was published by Ma et al. [24] using optimization on Riemannian manifolds [9].

The task of *direct calibration* provides a simple example illustrating the importance of a correct treatment of the statistical model selected. Since, in this paper, we are not concerned with the issue of outlier tolerance, we will assume that all the data points obey the model. Robust techniques, especially developed for the employed class of models, exist, e.g., [43], [4], and can be easily integrated into the proposed estimator.

In direct calibration, the relation between a 3D point $\boldsymbol{m}_{io}$ on a calibration grid and its 2D projection in the image plane, $\boldsymbol{z}_{io} = [z_{1io}\ z_{2io}]^\top$ (Fig. 1), is exploited to compute the intrinsic and extrinsic parameters of the camera [16, Chapter 6]. The ideal affine coordinates $\boldsymbol{M}_{io} = [\boldsymbol{m}_{io}^\top\ 1]^\top$ of the 3D points (not in the plane at infinity) are related to the ideal affine image coordinates $\boldsymbol{Z}_{io} = [\boldsymbol{z}_{io}^\top\ 1]^\top$ through P, an unknown 3×4 projection matrix

$$\boldsymbol{Z}_{io} \propto \mathrm{P}\boldsymbol{M}_{io} \quad i = 1, \ldots, n, \tag{6}$$

where $\propto$ means projective equivalence. The relation (6) can be rewritten as

$$\begin{bmatrix} \boldsymbol{M}_{io}^\top & \boldsymbol{0}^\top & -z_{1io}\boldsymbol{M}_{io}^\top \\ \boldsymbol{0}^\top & \boldsymbol{M}_{io}^\top & -z_{2io}\boldsymbol{M}_{io}^\top \end{bmatrix} \boldsymbol{p} = \mathrm{A}(\boldsymbol{z}_{io}, \boldsymbol{m}_{io})\boldsymbol{p} = \boldsymbol{0}, \tag{7}$$

where $\boldsymbol{p} = \mathrm{vec}(\mathrm{P}^\top)$ is the 12-dimensional column vector obtained by stacking up the rows of the matrix P. See the Appendix for the definition and properties of the *vec* operator.

In camera calibration, it is assumed that the available 3D coordinates of the calibration points $\boldsymbol{m}_{io}$ are accurate, while the image points $\boldsymbol{z}_i$ are measured with independent and identically distributed (i.i.d.) and zero-mean errors. The goal of the estimation process is to find $\hat{\boldsymbol{p}}$, the vector containing the elements of the projection matrix, from where the camera parameters can be derived.

The linear solution of the estimation problem is the simplest approach, sometimes used exclusively in textbooks, e.g., [39, Section 6.2]. It is also called the direct linear transformation (DLT) algorithm [16, p. 167] and is based on minimizing the algebraic errors. The algebraic error is computed by replacing the ideal image point coordinates $\boldsymbol{z}_{io}$ in (7) with the available measurements $\boldsymbol{z}_i$ and minimizing

$$\mathcal{J}(\hat{\boldsymbol{p}}) = \sum_{i=1}^{n} \frac{\hat{\boldsymbol{p}}^\top \mathrm{A}^\top(\boldsymbol{z}_i, \boldsymbol{m}_{io})\mathrm{A}(\boldsymbol{z}_i, \boldsymbol{m}_{io})\hat{\boldsymbol{p}}}{\|\hat{\boldsymbol{p}}\|^2}. \tag{8}$$

The normalization factor $\|\hat{\boldsymbol{p}}\|^2$ appearing at the denominator of each term is required to explicitly account for the gauge constraint and to avoid a zero trivial solution. The linear solution to (8) is the *total least squares* (TLS) estimate $\hat{\boldsymbol{p}}_{TLS}$, proportional to the "smallest" right singular vector (corresponding to the smallest singular value) of the $2n \times 12$ matrix

$$\mathrm{B} = \begin{bmatrix} \mathrm{A}(\boldsymbol{z}_1, \boldsymbol{m}_{1o}) \\ \vdots \\ \mathrm{A}(\boldsymbol{z}_n, \boldsymbol{m}_{no}) \end{bmatrix}. \tag{9}$$

A TLS estimate, however, is optimal only if the measurement noise associated with every row of the matrix B has the same covariance $\sigma^2\mathrm{I}$ [42, p. 227]. In the direct calibration task, this condition is violated since, in each pair of rows of B, the noisy image coordinates z_{1i} or z_{2i} are multiplied by a different vector $\boldsymbol{M}_{io}$, as seen in (7). For such a noise process, the TLS estimate becomes biased [28] and should only be used as the initial solution in a nonlinear estimation procedure minimizing geometric distances, i.e., the distance between a measured image point and the unknown projection of the corresponding 3D calibration point [16, p. 170]. We will return to the problem of camera calibration in Section 5.1.

We can conclude that, in spite of a linear dependence on the parameter $\boldsymbol{p}$, the model of camera calibration is more complex due to the presence of products between measured quantities. A similar situation arises in numerous computer vision tasks, in which information about the 3D environment is sought. Indeed, the fundamental property of projective geometry, the incidence relation, implies the presence of such products. The goal of this paper is to provide, in the context of computer vision problems, a unified approach for estimating models nonlinear, either in the measurements, in the parameter, or in both.

In Section 2, we present the linearization approach for solving EIV problems with separable parameters. In Section 3, the solution to the HEIV problem, together with statistical properties and numerical implementation, are discussed. The relationship of the HEIV estimator with other techniques proposed in the literature is analyzed in Section 4. Applications of the HEIV algorithm to 3D computer vision problems which translate in multivariate constraints are presented in Section 5. Finally, a discussion of other applications of the HEIV is given in Section 6.

2 An Approach through Linearization

The parameters of the separable EIV model are estimated by solving the following problem. Given that the noise-free, unobservable data $\boldsymbol{z}_{1o}, \ldots, \boldsymbol{z}_{no}$ and the unknown parameter vector $\boldsymbol{\beta}_o$ satisfy the constraint (3), find from the measurements $\boldsymbol{z}_i$, affected by additive heteroscedastic noise (2), the estimates $\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n$ and $\hat{\boldsymbol{\beta}}$ obeying

$$\boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\beta}}) = \Phi(\hat{\boldsymbol{z}}_i)\, \boldsymbol{\theta}(\hat{\boldsymbol{\beta}}) = \boldsymbol{0} \qquad \|\boldsymbol{\theta}(\hat{\boldsymbol{\beta}})\| = 1 \qquad i = 1, \ldots, n, \tag{10}$$

$$\Phi(\cdot) \in \mathbb{R}^{m \times p} \quad \boldsymbol{\theta}(\cdot) \in \mathbb{R}^p \quad \boldsymbol{\beta} \in \mathbb{R}^q.$$

The covariance matrices C_{z_i} of the noise are known, but the value of the constant σ_ν^2 is unknown. The constraint $\|\boldsymbol{\theta}(\hat{\boldsymbol{\beta}})\| = 1$ is required only to eliminate the scale ambiguity.

The estimates are obtained by minimizing the objective function

$$\mathcal{J}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n) = \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^\top \mathrm{C}_{z_i}^{+} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i), \tag{11}$$

i.e., the sum of squared Mahalanobis semidistances between a measurement and its unknown corrected value. To account for the case of rank deficient covariance matrices, the pseudoinverse denoted with the superscript "+" has to be used in (11). The common term σ_ν^2 can be dropped since it has no influence on the minimum of the objective function $\mathcal{J}$. Its estimate $\hat{\sigma}_\nu^2$ will be found after the main estimation procedure. The estimation process belongs to the class of maximum likelihood if the measurement noise is known to be normally distributed.

To account for the constraints (10), we will introduce the Lagrange multipliers $\boldsymbol{\eta}_i \in \mathbb{R}^m$ and seek to minimize

$$\mathcal{J}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n) = \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^\top \mathrm{C}_{z_i}^{+} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i) + \sum_{i=1}^{n} \boldsymbol{\eta}_i^\top \boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\beta}}). \tag{12}$$

We show next that by treating the separable EIV model (10) as linear in the parameter $\boldsymbol{\theta}$, an efficient minimization of (12) can be done by iteratively solving a generalized eigenvalue problem. The most general case of solving explicitly in the parameter $\boldsymbol{\beta}$ is addressed in Section 3.3.

Our goal is to obtain the estimates $\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n$ and $\hat{\boldsymbol{\theta}}$ by minimizing the objective function $\mathcal{J}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n)$ obtained from (12) by substituting $\hat{\boldsymbol{\theta}}$ for $\hat{\boldsymbol{\beta}}$ and letting $\boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}}) = \Phi(\boldsymbol{z}_i)\hat{\boldsymbol{\theta}}$. Note that, since we did not require the parameter $\hat{\boldsymbol{\theta}}$ to be a unit vector, a gauge freedom up to multiplication with a nonzero constant is present.

The main and nuisance parameter estimates must obey

$$\boldsymbol{J}_{\mathcal{J}|\hat{\theta}}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n) = \boldsymbol{0} \tag{13}$$

$$\boldsymbol{J}_{\mathcal{J}|\hat{\boldsymbol{z}}_i}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n) = \boldsymbol{0} \quad i = 1, \ldots, n, \tag{14}$$

where $\boldsymbol{J}_{\mathcal{J}|u}$ denotes the gradient of the scalar objective function with respect the vector $\boldsymbol{u}$.

From (14), using vector calculus chain rules (A.17), (A.18), we obtain

$$\mathrm{C}_{z_i}^{+}(\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i) = \mathrm{J}_{f|\hat{z}_i}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i. \tag{15}$$

Note that a singular covariance matrix C_{z_i} implies that some of the elements of the measurement vector $\boldsymbol{z}_i$ may not be corrupted by noise. For these elements, the measured value and an unbiased estimate are identical, i.e., their difference is zero. This observation allows us to express (15) after simple manipulations with an orthogonal projector into the range of the covariance matrix [14, p. 75] as

$$\hat{\boldsymbol{z}}_i = \boldsymbol{z}_i - \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i. \tag{16}$$

Let $\check{\boldsymbol{z}}_i$ be an available estimate of $\boldsymbol{z}_{io}$. Initially, in the absence of any prior information, we can assume that $\check{\boldsymbol{z}}_i = \boldsymbol{z}_i$. In the following, we use $\check{\boldsymbol{z}}_i$ to discriminate between an available estimate of the unknown ideal point $\boldsymbol{z}_{io}$ and the measured value $\boldsymbol{z}_i$. The first order Taylor expansion of $\boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})$ around $\check{\boldsymbol{z}}_i$, for a given $\hat{\boldsymbol{\theta}}$, yields

$$\boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = \boldsymbol{f}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) + \mathrm{J}_{f|\hat{z}_i}(\check{z}_i, \hat{\boldsymbol{\theta}})^\top (\hat{\boldsymbol{z}}_i - \check{\boldsymbol{z}}_i). \tag{17}$$

We will now make the assumption that the Jacobian matrix does not change significantly when computed in $\check{\boldsymbol{z}}_i$ instead of $\hat{\boldsymbol{z}}_i$. Then, (16) can be written as

$$\hat{\boldsymbol{z}}_i - \check{\boldsymbol{z}}_i = \boldsymbol{z}_i - \check{\boldsymbol{z}}_i - \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\check{z}_i, \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i. \tag{18}$$

From (17), we obtain

$$\begin{aligned} \boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) - \boldsymbol{f}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = {} & \mathrm{J}_{f|\hat{\boldsymbol{z}}_i}(\check{z}_i, \hat{\boldsymbol{\theta}})^\top (\boldsymbol{z}_i - \check{\boldsymbol{z}}_i) \\ & - \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^\top \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i \end{aligned} \tag{19}$$

and, taking into account (10), we have

$$\boldsymbol{f}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) + \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^\top (\boldsymbol{z}_i - \check{\boldsymbol{z}}_i) = \left[\mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^\top \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\right] \boldsymbol{\eta}_i. \tag{20}$$

In a first order approximation, the left-hand side of (20) is the Taylor expansion of $\boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}})$ around $\check{\boldsymbol{z}}_i$, hence, the Lagrange multipliers can be expressed as

$$\boldsymbol{\eta}_i = \left[\mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^\top \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\right]^{+} \boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}}). \tag{21}$$

One can recognize in the brackets the first order approximation of the covariance of the expression $\boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}})$, i.e., the result of error propagation from $\boldsymbol{z}_i$ to $\boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}})$, computed in $\check{\boldsymbol{z}}_i$

$$\mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) \triangleq \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^\top \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}). \tag{22}$$

The rank $r \leq m$ of the covariance of the constraint $\mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})$ is called the rank of the constraint. When $r < m$, the constraint is called singular [20, pp. 131-133]. Thus, the estimate of the ideal value $\boldsymbol{z}_{io}$ is obtained from (16) as

$$\hat{\boldsymbol{z}}_i = \boldsymbol{z}_i - \mathrm{C}_{z_i}\mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^+ \boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}}). \tag{23}$$

Note that (23) represents the first order approximation of the projection of the measurement $\boldsymbol{z}_i$ onto the manifold defined by the multivariate constraint $\boldsymbol{f}(\boldsymbol{z}, \hat{\boldsymbol{\theta}})$ under the metric induced by the covariance matrix C_{z_i}. Using (18), after some simple manipulations, the objective function (11) can be an expressed function of the previous estimate of the corrected measurement $\check{\boldsymbol{z}}_i$ and the current parameter estimate $\hat{\boldsymbol{\theta}}$ as

$$\mathcal{J}_{HEIV}(\hat{\boldsymbol{\theta}}) = \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}})^\top \mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^+ \boldsymbol{f}(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}}). \tag{24}$$

Assuming that $\check{\boldsymbol{z}}_i = \boldsymbol{z}_i$, we can express the cost function (24), only function of the noisy measurements $\boldsymbol{z}_i$ and the value of the parameter estimate $\hat{\boldsymbol{\theta}}$.

Expression (24) is a sum of squared Mahalanobis semi-distances, where each term is associated with the corresponding error propagated covariance matrix. In the particular case of univariate constraints, $f(\boldsymbol{z}_i, \hat{\boldsymbol{\theta}})$ becomes the algebraic distance from the solution surface.

We have

$$\begin{aligned}\mathrm{J}_{\mathcal{J}|\hat{\theta}}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n) &= \sum_{i=1}^{n} \mathrm{J}_{f|\hat{\theta}}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i = \sum_{i=1}^{n} \Phi(\hat{\boldsymbol{z}}_i)^\top \boldsymbol{\eta}_i \\ &= \sum_{i=1}^{n} (\boldsymbol{\eta}_i^\top \otimes \mathrm{I}_p)\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i) = \boldsymbol{0},\end{aligned} \tag{25}$$

where $\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i) = \mathrm{vec}(\Phi(\hat{\boldsymbol{z}}_i)^\top)$ with the vectorization operation $\mathrm{vec}(\cdot)$ and the Kronecker product $\otimes$ being defined in the Appendix. Relation (25) is obtained by applying (A.13) for the $p \times m$ matrix $\Phi^\top$ and the m-dimensional vector $\boldsymbol{\eta}_i$. The first order expansion of $\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i)$ around $\check{\boldsymbol{z}}_i$ yields

$$\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i) = \boldsymbol{\varphi}(\check{\boldsymbol{z}}_i) + \mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i)^\top(\hat{\boldsymbol{z}}_i - \check{\boldsymbol{z}}_i) \tag{26}$$

which can be written using (18) as

$$\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i) = \boldsymbol{\varphi}(\check{\boldsymbol{z}}_i) + \mathrm{J}_{\varphi|\hat{\boldsymbol{z}}_i}(\check{\boldsymbol{z}}_i)^\top[\boldsymbol{z}_i - \check{\boldsymbol{z}}_i - \mathrm{C}_{z_i}\mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i]. \tag{27}$$

Recalling that $\boldsymbol{\varphi}(\check{\boldsymbol{z}}_i)$ is the vector obtained by stacking up the rows of the matrix $\Phi(\check{\boldsymbol{z}}_i)$ and the definition of the Kronecker product (A.2), we have

$$\boldsymbol{f}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = \Phi(\check{\boldsymbol{z}}_i)\hat{\boldsymbol{\theta}} = (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}})^\top \boldsymbol{\varphi}(\check{\boldsymbol{z}}_i) \tag{28}$$

from where, after applying the Jacobian (A.2) results,

$$\mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = \mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i)(\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}}) \tag{29}$$

and, therefore,

$$\begin{aligned}\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i) = \boldsymbol{\varphi}(\check{\boldsymbol{z}}_i) &+ \mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i)^\top(\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i) \\ &- \mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i)^\top \mathrm{C}_{z_i}\mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i)(\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}})\boldsymbol{\eta}_i.\end{aligned} \tag{30}$$

The covariance of $\boldsymbol{\varphi}(\boldsymbol{z}_i)$ approximated through error propagation in $\check{\boldsymbol{z}}_i$ is

$$\mathrm{C}_\varphi(\check{\boldsymbol{z}}_i) \triangleq \mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i)^\top \mathrm{C}_{z_i}\mathrm{J}_{\varphi|\hat{z}_i}(\check{\boldsymbol{z}}_i) \tag{31}$$

and, after recognizing in the first two terms the linear approximation of $\boldsymbol{\varphi}(\boldsymbol{z}_i)$ around $\check{\boldsymbol{z}}_i$, and using (A.13) for the vectors $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\eta}_i^\top$, (30) becomes

$$\boldsymbol{\varphi}(\hat{\boldsymbol{z}}_i) = \boldsymbol{\varphi}(\boldsymbol{z}_i) - \mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)(\boldsymbol{\eta}_i \otimes \mathrm{I}_p)\hat{\boldsymbol{\theta}}. \tag{32}$$

Since the Jacobian $\mathrm{J}_{\mathcal{J}|\hat{\theta}}$ must vanish at the solution, it follows from (32) that (25) can be written as

$$\sum_{i=1}^{n} (\boldsymbol{\eta}_i^\top \otimes \mathrm{I}_p)\boldsymbol{\varphi}(\boldsymbol{z}_i) = \sum_{i=1}^{n} (\boldsymbol{\eta}_i \otimes \mathrm{I}_p)^\top \mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)(\boldsymbol{\eta}_i \otimes \mathrm{I}_p)\hat{\boldsymbol{\theta}}, \tag{33}$$

where we made use of (A.4). The left side of (33) can be rewritten after taking into account (21) and (22) as

$$\begin{aligned}\sum_{i=1}^{n}(\boldsymbol{\eta}_i^\top \otimes \mathrm{I}_p)\boldsymbol{\varphi}(\boldsymbol{z}_i) &= \sum_{i=1}^{n} \Phi(\boldsymbol{z}_i)^\top \boldsymbol{\eta}_i \\ &= \sum_{i=1}^{n} \Phi(\boldsymbol{z}_i)^\top \mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^+ \Phi(\boldsymbol{z}_i)\hat{\boldsymbol{\theta}}.\end{aligned} \tag{34}$$

The gradient of the cost function with respect to $\hat{\boldsymbol{\theta}}$ from (25) can finally be written as

$$\mathrm{J}_{\mathcal{J}|\hat{\theta}}(\hat{\boldsymbol{\theta}}) = [\mathrm{S}(\hat{\boldsymbol{\theta}}) - \mathrm{C}(\hat{\boldsymbol{\theta}})]\,\hat{\boldsymbol{\theta}} \triangleq \mathrm{N}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} = \boldsymbol{0}, \tag{35}$$

where the *weighted scatter matrix* $\mathrm{S}(\hat{\boldsymbol{\theta}})$ is

$$\mathrm{S}(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \Phi(\boldsymbol{z}_i)^\top \mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})^+ \Phi(\boldsymbol{z}_i) \tag{36}$$

and the *weighted covariance* matrix $\mathrm{C}(\hat{\boldsymbol{\theta}})$ is

$$\mathrm{C}(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} (\boldsymbol{\eta}_i \otimes \mathrm{I}_p)^\top \mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)(\boldsymbol{\eta}_i \otimes \mathrm{I}_p). \tag{37}$$

Note that several first order approximations were employed while deriving the expressions of the matrices $\mathrm{S}(\boldsymbol{\theta})$ and $\mathrm{C}(\boldsymbol{\theta})$. All these approximations were linearizations around $\check{\boldsymbol{z}}_i$, the *previous* value of the corrected measurements, and $\hat{\boldsymbol{\theta}}$, the *current* value of the parameter estimate.

Since both matrices $\mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})$ and $\mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)$ are positive semidefinite, $\mathrm{S}(\hat{\boldsymbol{\theta}})$ and $\mathrm{C}(\hat{\boldsymbol{\theta}})$ are also positive semidefinite. The connection between the covariance matrices $\mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})$ and $\mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)$ is obtained by substituting (29) into (22)

$$\mathrm{C}_f(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}})^\top \mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)(\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}}). \tag{38}$$

Using the expression of the constraint and (36), the objective function (24) can be written as

$$\mathcal{J}(\hat{\boldsymbol{\theta}}) = \frac{1}{2}\hat{\boldsymbol{\theta}}^\top\, \mathrm{S}(\hat{\boldsymbol{\theta}})\, \hat{\boldsymbol{\theta}}. \tag{39}$$

Solving (35) can be expressed as a generalized eigenproblem type equation

$$\mathrm{S}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} = \mathrm{C}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} \tag{40}$$

called the *heteroscedastic errors-in-variables* (HEIV) equation. The gauge freedom up to multiplication of $\hat{\boldsymbol{\theta}}$ with a nonzero constant is preserved due to the special structure of the two matrices, since, for any $a \neq 0$, $\mathrm{S}(a\hat{\boldsymbol{\theta}}) = a^{-2}\mathrm{S}(\hat{\boldsymbol{\theta}})$ and $\mathrm{C}(a\hat{\boldsymbol{\theta}}) = a^{-2}\mathrm{C}(\hat{\boldsymbol{\theta}})$.

The HEIV equation (40) generalizes for multivariate constraints with a separable parameter the scalar case described in [23], which is also equivalent to the fundamental numerical scheme (FNS) of Chojnacki et al. [7]. Being the most general expression for solving separable EIV models linear in the parameter $\boldsymbol{\theta}$, the HEIV equation plays a central role in many computer vision problems. In Section 4,

we will show that different particular cases of (40) appear in several techniques used in computer vision: the generalized total least squares (GTLS) [18], the Sampson method [35], [16, p. 97], and the renormalization method of Kanatani [20].

3 Solving the HEIV Equation

The HEIV equation (40) is nonlinear since the unknown $\hat{\boldsymbol{\theta}}$ also appears in the matrices $\mathrm{S}(\hat{\boldsymbol{\theta}})$ and $\mathrm{C}(\hat{\boldsymbol{\theta}})$. An iterative solution can be found by iteratively solving the generalized eigenvalue problem

$$\mathrm{S}(\check{\boldsymbol{\theta}})\,\hat{\boldsymbol{\theta}} = \lambda \mathrm{C}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} \tag{41}$$

and, at each step, the already available *previous* estimate of the parameter $\check{\boldsymbol{\theta}}$ is changed to the current value $\hat{\boldsymbol{\theta}}$ obtained as the smallest generalized eigenvector of (41), i.e., the generalized eigenvector corresponding to the smallest generalized eigenvalue. Such an estimation technique was proposed in an equivalent form by Fuller [12, p. 217].

The Rayleigh quotient

$$\lambda(\boldsymbol{\theta}) = \frac{\boldsymbol{\theta}^\top \mathrm{S}(\check{\boldsymbol{\theta}})\boldsymbol{\theta}}{\boldsymbol{\theta}^\top \mathrm{C}(\check{\boldsymbol{\theta}})\boldsymbol{\theta}} \tag{42}$$

satisfies [14, p. 465]

$$\lambda(\boldsymbol{\theta}) = \operatorname*{argmin}_{\lambda} \|\mathrm{S}(\check{\boldsymbol{\theta}})\boldsymbol{\theta} - \lambda\,\mathrm{C}(\check{\boldsymbol{\theta}})\boldsymbol{\theta}\|_{\mathrm{C}}^2, \tag{43}$$

where $\|\boldsymbol{\theta}\|_{\mathrm{C}}^2$ is the squared Mahalanobis norm defined as $\|\boldsymbol{\theta}\|_C^2 = \boldsymbol{\theta}^\top \mathrm{C}^+\boldsymbol{\theta}$. The smallest generalized eigenvalue λ_{min} of (41) satisfies $\lambda_{min} = \min_{\boldsymbol{\theta}} \lambda(\boldsymbol{\theta})$ and, thus, the corresponding eigenvector is the solution of the update equation.

The matrices $\mathrm{S}(\check{\boldsymbol{\theta}})$ and $\mathrm{C}(\check{\boldsymbol{\theta}})$ are symmetric and positive semidefinite and, thus, the eigenvalues of (41) are nonnegative, i.e., $\lambda_{min} \geq 0$. We restrict ourselves for the moment to positive definite $\mathrm{S}(\check{\boldsymbol{\theta}})$ matrices, i.e., $\check{\boldsymbol{\theta}}^\top \mathrm{S}(\check{\boldsymbol{\theta}})\check{\boldsymbol{\theta}} \neq 0$. Then, from (B.1) at every iteration, we have

$$\lambda(\check{\theta}) = \frac{\check{\boldsymbol{\theta}}^\top \mathrm{S}(\check{\boldsymbol{\theta}})\check{\boldsymbol{\theta}}}{\check{\boldsymbol{\theta}}^\top \mathrm{C}(\check{\boldsymbol{\theta}})\check{\boldsymbol{\theta}}} = 1 \tag{44}$$

and, thus, $0 < \lambda_{min} \leq 1$. At convergence, $\check{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}$ up to the allowed tolerance, which implies $\lambda_{min} = 1$, thus $\mathrm{S}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} = \mathrm{C}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}$. A similar argument was used in [20, p. 283], though in a somewhat different context.

The second order Taylor expansion of the objective function (39) around $\check{\boldsymbol{\theta}}$ yields

$$\begin{aligned}\mathcal{J}(\hat{\boldsymbol{\theta}}) &= \mathcal{J}(\check{\boldsymbol{\theta}}) + \mathrm{J}_{\mathcal{J}|\hat{\theta}}(\check{\boldsymbol{\theta}})^\top(\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}}) + \frac{1}{2}(\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}})^\top \frac{\partial^2 \mathcal{J}(\check{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}}) \\ &= \mathcal{J}(\check{\boldsymbol{\theta}}) + (\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}})^\top \mathrm{N}(\check{\boldsymbol{\theta}})\check{\boldsymbol{\theta}} + \frac{1}{2}(\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}})^\top \mathrm{N}(\check{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}}) \\ &= \mathcal{J}(\check{\boldsymbol{\theta}}) + \frac{1}{2}\hat{\boldsymbol{\theta}}^\top \mathrm{N}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}},\end{aligned} \tag{45}$$

where we used the expression of the gradient (35) computed for $\check{\boldsymbol{\theta}}$. From this expression, the Hessian matrix was approximated as $\mathrm{N}(\check{\boldsymbol{\theta}})$ by *disregarding* the dependence of $\mathrm{S}(\check{\boldsymbol{\theta}})$ and $\mathrm{C}(\check{\boldsymbol{\theta}})$ on $\check{\boldsymbol{\theta}}$. We also took into account that $\check{\boldsymbol{\theta}}^\top \mathrm{N}(\check{\boldsymbol{\theta}})\check{\boldsymbol{\theta}} = 0$. Then,

$$\begin{aligned}\mathcal{J}(\hat{\boldsymbol{\theta}}) - \mathcal{J}(\check{\boldsymbol{\theta}}) &= \frac{1}{2}\hat{\boldsymbol{\theta}}^\top \left[\mathrm{S}(\check{\boldsymbol{\theta}}) - \mathrm{C}(\check{\boldsymbol{\theta}})\right]\hat{\boldsymbol{\theta}} \\ &= \frac{1}{2}\hat{\boldsymbol{\theta}}^\top \left[\lambda_{min}\mathrm{C}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} - \mathrm{C}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}\right] \\ &= \frac{1}{2}(\lambda_{min} - 1)\hat{\boldsymbol{\theta}}^\top \mathrm{C}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} \leq 0\end{aligned} \tag{46}$$

since $\mathrm{C}(\check{\boldsymbol{\theta}})$ is a positive semidefinite matrix. We conclude that the value of the objective function decreases with the iterations and convergence is reached when the smallest eigenvalue of the generalized eigenproblem (41) becomes one.

Note that, during the HEIV update equation, we select the smallest generalized eigenvector at each step, as opposed to the one which is closest to one. This choice was experimentally validated through numerous simulations. An insight can be obtained from (46) which shows that the cost function decrease is highest when $\lambda_{min} - 1$ is the smallest. Similarly, the HEIV is shown in Section 4.1 to generalize the GTLS algorithm which also employs the smallest generalized eigenvalue of a similar eigenproblem.

There is an exception to this convergence rule, that of uncorrupted data. Then, the weighted covariance matrix $\mathrm{C}(\check{\boldsymbol{\theta}})$ vanishes and the weighted scatter matrix $\mathrm{S}_o(\check{\boldsymbol{\theta}})$ becomes singular, where the subscript "o" makes explicit the fact that the scatter matrix is evaluated at the noise-free values of the measurements. Thus, we have $\mathrm{S}_o(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} = \mathbf{0}$ and the estimate $\hat{\boldsymbol{\theta}}$ is the eigenvector corresponding to the eigenvalue $\lambda = 0$ for any $\check{\boldsymbol{\theta}} \in \mathbb{R}^p$.

In our implementation of the algorithm, we detect whether the TLS estimate yields very small residuals, i.e., the noise of the data is very small. In this case, any linear estimate TLS, GTLS, and HEIV will yield the same solution. This is a singularity of the algorithm in which the Rayleigh quotient converges to zero instead of one.

In all the experiments, convergence was reached in a few (typically, two or three) iterations. The iterations toward solving the HEIV equation have some similarity (though, are certainly not identical) to the method of Rayleigh quotient for solving generalized symmetric eigenproblems. The latter is known to have cubic convergence [14, Section 8.2.4] related to the fast convergence of our method.

3.1 Implementation Using Generalized Singular Value Decomposition

A numerically robust scheme to solve (41) is based on the generalized singular value decomposition (GSVD) [14, Section 8.7.3]. Leedan and Meer [23] solve for the case of scalar constraints, where a more detailed description of GSVD can be found.

The square root of the positive semidefinite matrix $\mathrm{C}_f(\hat{\boldsymbol{z}}_i, \check{\boldsymbol{\theta}})^+$ (22) can be computed with regular SVD which, though more computationally intensive than the Cholesky decomposition, has the advantage of handling singular matrices. The square root is the matrix $\Gamma(\check{\boldsymbol{z}}_i, \check{\boldsymbol{\theta}}) \in \mathbb{R}^{m\times m}$ such that

$$\mathrm{C}_f(\check{\boldsymbol{z}}_i, \check{\boldsymbol{\theta}})^+ = \Gamma(\check{\boldsymbol{z}}_i, \check{\boldsymbol{\theta}})^\top \Gamma(\check{\boldsymbol{z}}_i, \check{\boldsymbol{\theta}}). \tag{47}$$

The matrix $\mathrm{K}(\check{\boldsymbol{\theta}}) \in \mathbb{R}^{mn\times p}$ is defined as

$$\mathrm{K}(\check{\boldsymbol{\theta}}) = \begin{bmatrix} \Gamma(\check{\boldsymbol{z}}_1, \check{\boldsymbol{\theta}})\Phi(\boldsymbol{z}_1) \\ \vdots \\ \Gamma(\check{\boldsymbol{z}}_n, \check{\boldsymbol{\theta}})\Phi(\boldsymbol{z}_n) \end{bmatrix} \tag{48}$$

and, from (36), results

$$\mathrm{S}(\check{\boldsymbol{\theta}}) = \mathrm{K}(\check{\boldsymbol{\theta}})^\top \mathrm{K}(\check{\boldsymbol{\theta}}). \tag{49}$$

Similarly, the square root of $\mathrm{C}(\check{\boldsymbol{\theta}})$ (37) is the matrix $\mathrm{L}(\check{\boldsymbol{\theta}}) \in \mathbb{R}^{p\times p}$ such that

$$\mathrm{C}(\check{\boldsymbol{\theta}}) = \mathrm{L}(\check{\boldsymbol{\theta}})^\top \mathrm{L}(\check{\boldsymbol{\theta}}). \tag{50}$$

The GSVD of the matrix pair $\mathrm{K}(\check{\boldsymbol{\theta}})$ and $\mathrm{L}(\check{\boldsymbol{\theta}})$ is obtained by simultaneous diagonalization with the help of two orthogonal matrices $\mathrm{U} \in \mathbb{R}^{mn\times mn}$, $\mathrm{V} \in \mathbb{R}^{p\times p}$, and a nonsingular matrix $\mathrm{Y} \in \mathbb{R}^{p\times p}$. That is,

$$\mathrm{U}^\top \mathrm{K}(\check{\boldsymbol{\theta}})\mathrm{Y} = \mathrm{diag}(d_{K_1}, \ldots, d_{K_p}) \in \mathbb{R}^{mn\times p} \quad d_{K_i} \geq d_{K_{i+1}} \geq 0, \tag{51}$$

$$\mathrm{V}^\top \mathrm{L}(\check{\boldsymbol{\theta}})\mathrm{Y} = \mathrm{diag}(d_{L_1}, \ldots, d_{L_p}) \in \mathbb{R}^{p\times p} \quad 0 \leq d_{L_i} \leq d_{L_{i+1}}, \tag{52}$$

where $d_{K_i}^2 + d_{L_i}^2 = 1$, $i = 1, \ldots, p$. The generalized singular values are defined as

$$\sigma_i = \frac{d_{K_i}}{d_{L_i}} \quad i = 1, \ldots, p \tag{53}$$

and the corresponding generalized singular vector is $\boldsymbol{y}_i$, the ith column of Y. The solution $\hat{\boldsymbol{\theta}}$ of (41) is the generalized singular vector corresponding to the smallest generalized singular value.

The use of GSVD instead of eigendecomposition has several advantages. It can handle the rank deficiency of the involved matrices, the square root matrices are better conditioned than their source, and, thus, the GSVD assures a better numerical behavior for the iterative procedure.

3.2 Statistical Properties

To assess the accuracy of the HEIV estimates, the first two moments can be computed in first order approximation. In [25], it is shown that the HEIV estimate is consistent for linear functions $\Phi(\boldsymbol{z}_i)$ and heteroscedastic measurements and that the parameter $\hat{\boldsymbol{\theta}}$ is unbiased in the first order approximation.

The theoretical covariance of the parameter $\hat{\boldsymbol{\theta}}$ is

$$\mathrm{C}_{\hat{\theta}} = \sigma_\nu^2 \mathrm{S}_o(\boldsymbol{\theta}_o)^+ \quad \|\boldsymbol{\theta}_o\| = 1. \tag{54}$$

Note that (54) requires that the gauge freedom of the parameter must be removed by imposing the norm one on $\boldsymbol{\theta}_o$. We have used the pseudoinverse of the scatter because, for uncorrupted data, the scatter is singular, having a rank $p-1$. It can be shown [25] that an estimate of $\mathrm{C}_{\hat{\theta}}$, after the convergence of the algorithm, is

$$\hat{\mathrm{C}}_{\hat{\theta}} = \hat{\sigma}_\nu^2 \Big[\mathrm{S}(\hat{\boldsymbol{\theta}}) - \lambda_{min}\mathrm{C}(\hat{\boldsymbol{\theta}})\Big]^+. \tag{55}$$

The estimate $\hat{\sigma}_\nu^2$ of the noise variance σ_ν^2 (for the proof, see [25]) is

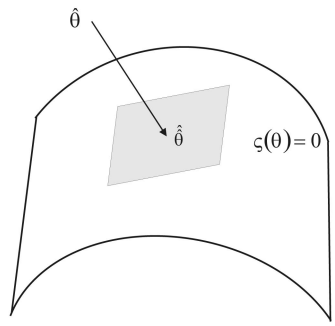


Fig. 2. The additional constraints are obtained by projecting the solution $\hat{\boldsymbol{\theta}}$ onto the manifold $\boldsymbol{\zeta}(\boldsymbol{\theta}) = \mathbf{0}$ in the metric induced by the covariance of the parameter, $\mathrm{C}_{\hat{\theta}}$.

$$\hat{\sigma}_\nu^2 = \frac{\hat{\boldsymbol{\theta}}^\top \mathrm{S}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}}{nr - p + 1}, \tag{56}$$

where $r \leq m$ is the rank of the covariance of the constraint $\boldsymbol{f}(\boldsymbol{z}_i, \boldsymbol{\theta})$ expressed in (22).

Assuming that the noise affecting the measurements has zero mean and $\hat{\boldsymbol{\theta}}$ is unbiased in the first order approximation, the estimates $\hat{\boldsymbol{z}}_i$ are also unbiased in the first order approximation [25].

3.3 Handling Additional Constraints

The parameter estimate may be required to satisfy an additional constraint

$$\boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}) = \mathbf{0}, \quad \boldsymbol{\zeta}(\,\cdot\,) \in \mathbb{R}^t. \tag{57}$$

For example, the fundamental matrix must have rank two, the ellipse must be a positive definite form, etc. The nonlinear additional constraints $\boldsymbol{\zeta}(\,\cdot\,)$ usually preclude the use of (generalized) eigenproblems in directly finding the solution $\hat{\boldsymbol{\theta}}$ which satisfies (10) and (57) simultaneously. A special case of scalar additional constraints, homogeneous of degree κ, $\zeta(t\hat{\boldsymbol{\theta}}) = t^\kappa \zeta(\hat{\boldsymbol{\theta}})$, was explored in the CFNS algorithm [41], [8], which showed that the constraint can be elegantly included in the optimization scheme.

The approach proposed in the paper is generally applicable to multivariate constraints which are not necessarily homogeneous in the parameter $\boldsymbol{\theta}$. We first find an estimate $\hat{\boldsymbol{\theta}}$ by imposing no other constraints than (10) and then force the additional constraint (57) to obtain $\hat{\hat{\boldsymbol{\theta}}}$ which obeys $\boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}) = \mathbf{0}$. The additional constraints are imposed by projecting the $\hat{\boldsymbol{\theta}}$ onto the manifold (59) under the metric induced by $\mathrm{C}_{\hat{\theta}}$, as shown in Fig. 2. A similar solution was proposed in [20, pp. 286-289].

Formally,

$$\hat{\hat{\boldsymbol{\theta}}} = \arg\min_{\theta} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^\top \mathrm{C}_{\hat{\theta}}^+ (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \tag{58}$$

subject to

$$\boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}) = \mathbf{0}. \tag{59}$$

Introducing the Lagrange multiplier $\boldsymbol{\eta} \in \mathbb{R}^t$, the solution $\hat{\hat{\boldsymbol{\theta}}}$ is found from

$$\frac{\partial \mathcal{J}}{\partial \hat{\hat{\boldsymbol{\theta}}}} = \mathbf{0} \quad \mathcal{J} = \frac{1}{2}\left(\hat{\boldsymbol{\theta}} - \hat{\hat{\boldsymbol{\theta}}}\right)^\top \mathrm{C}_{\hat{\theta}}^{+}\left(\hat{\boldsymbol{\theta}} - \hat{\hat{\boldsymbol{\theta}}}\right) + \boldsymbol{\eta}^\top \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}) \tag{60}$$

from where we obtain

$$\hat{\hat{\boldsymbol{\theta}}} = \hat{\boldsymbol{\theta}} - \mathrm{C}_{\hat{\theta}} \frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}} \boldsymbol{\eta}. \tag{61}$$

A first order Taylor expansion yields

$$\boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}) = \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}) + \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right]^\top \left(\hat{\boldsymbol{\theta}} - \hat{\hat{\boldsymbol{\theta}}}\right) \tag{62}$$

or

$$\boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}) = \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right]^\top \mathrm{C}_{\hat{\theta}} \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right] \boldsymbol{\eta}, \tag{63}$$

thus

$$\boldsymbol{\eta} = \left\{\left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right]^\top \mathrm{C}_{\hat{\theta}} \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right]\right\}^{+} \boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}). \tag{64}$$

The solution sought is

$$\hat{\hat{\boldsymbol{\theta}}} = \hat{\boldsymbol{\theta}} - \mathrm{C}_{\hat{\theta}} \frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}} \left\{\left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right]^\top \mathrm{C}_{\hat{\theta}} \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}}\right]\right\}^{+} \boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}). \tag{65}$$

Equation (65) doesn't have a close form solution; thus, the following update may be employed:

$$\hat{\hat{\boldsymbol{\theta}}}^{[j+1]} = \hat{\hat{\boldsymbol{\theta}}}^{[j]} - \mathrm{C}_{\hat{\theta}} \frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}[j])^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}^{[j]}} \left\{\left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}^{[j]}}\right]^\top \mathrm{C}_{\hat{\theta}} \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}^{[j]}}\right]\right\}^{+} \boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}^{[j]}). \tag{66}$$

Since

$$\mathrm{C}_{\hat{\theta}} \propto \mathrm{N}_\lambda(\hat{\boldsymbol{\theta}})^{+} \quad \mathrm{N}_\lambda(\hat{\boldsymbol{\theta}}) = \mathrm{S}(\hat{\boldsymbol{\theta}}) - \lambda_{min}\mathrm{C}(\hat{\boldsymbol{\theta}}), \tag{67}$$

(66) can be finally written as

$$\begin{aligned}\hat{\hat{\boldsymbol{\theta}}}^{[j+1]} &= \hat{\hat{\boldsymbol{\theta}}}^{[j]} - \mathrm{N}_\lambda(\hat{\boldsymbol{\theta}})^{+} \frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}^{[j]}} \\ &\left\{\left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}^{[j]}}\right]^\top \mathrm{N}_\lambda(\hat{\boldsymbol{\theta}})^{+} \left[\frac{\partial \boldsymbol{\zeta}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})^\top}{\partial \hat{\hat{\boldsymbol{\theta}}}^{[j]}}\right]\right\}^{+} \boldsymbol{\zeta}(\hat{\boldsymbol{\theta}}^{[j]}).\end{aligned} \tag{68}$$

Equation (68) must be iterated two or three times until (59) is satisfied since the manifold $\boldsymbol{\zeta}(\cdot)$ was approximated locally by a hyperplane. Enforcing the ancillary constraints can be incorporated into the main HEIV algorithm summarized in Section 3.5 such that, at each HEIV iteration step, the estimate satisfies (57).

3.4 Nonlinear EIV Model with Separable Parameter

In this section, we present a solution for solving the nonlinear EIV model (4) in its most general form when there is a nonlinear dependency on the parameter $\boldsymbol{\beta}_o$. In the linearized algorithms, a solution can be obtained by finding the minimum of the cost function in the linearized space (i.e., on the manifold $\|\hat{\boldsymbol{\theta}}\| = 1$), followed by estimating the solution $\hat{\boldsymbol{\beta}}$ which best approximates $\boldsymbol{\theta}(\hat{\boldsymbol{\beta}}) \approx \hat{\boldsymbol{\theta}}$. However, in some cases, minimizing the unconstrained objective function can lead to convergence to the incorrect solution as shown by Matei et al. in the case of trifocal tensor estimation [26]. Instead, the HEIV algorithm derived in Section 2 can be extended to allow enforcing the constraint $\boldsymbol{\theta}(\hat{\boldsymbol{\beta}}) \approx \hat{\boldsymbol{\theta}}$ at each iteration of the HEIV algorithm.

Expressing (39) in the parameter $\hat{\boldsymbol{\beta}}$, the cost function which we want to minimize is

$$\mathcal{J}_{HEIV}(\hat{\boldsymbol{\beta}}) = \frac{1}{2}\boldsymbol{\theta}(\hat{\boldsymbol{\beta}})^\top \mathrm{S}(\hat{\boldsymbol{\beta}})\boldsymbol{\theta}(\hat{\boldsymbol{\beta}}), \tag{69}$$

where $\mathrm{S}(\hat{\boldsymbol{\beta}}) = \mathrm{S}(\boldsymbol{\theta}(\hat{\boldsymbol{\beta}}))$ is the weighted scatter matrix defined in (36). Using the chain rule, the gradient of (69) must satisfy

$$\frac{\partial \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})^\top}{\partial \hat{\boldsymbol{\beta}}}\left[\mathrm{S}(\hat{\boldsymbol{\beta}}) - \mathrm{C}(\hat{\boldsymbol{\beta}})\right]\boldsymbol{\theta}(\hat{\boldsymbol{\beta}}) = \mathbf{0}. \tag{70}$$

Using the Taylor expansion of $\boldsymbol{\theta}(\boldsymbol{\beta})$ around the current estimate $\check{\boldsymbol{\beta}}$,

$$\boldsymbol{\theta}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\theta}(\check{\boldsymbol{\beta}}) + \frac{\partial \boldsymbol{\theta}(\check{\boldsymbol{\beta}})}{\partial \check{\boldsymbol{\beta}}^\top}(\hat{\boldsymbol{\beta}} - \check{\boldsymbol{\beta}}), \tag{71}$$

and, by combining (70) and (71), the update equation becomes

$$\begin{aligned}\hat{\boldsymbol{\beta}} - \check{\boldsymbol{\beta}} &= -\left[\frac{\partial \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})^\top}{\partial \hat{\boldsymbol{\beta}}}\mathrm{N}(\hat{\boldsymbol{\beta}})\frac{\partial \boldsymbol{\theta}(\check{\boldsymbol{\beta}})}{\partial \check{\boldsymbol{\beta}}^\top}\right]^{+} \frac{\partial \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})^\top}{\partial \hat{\boldsymbol{\beta}}}\mathrm{N}(\hat{\boldsymbol{\beta}})\boldsymbol{\theta}(\check{\boldsymbol{\beta}}) \\ \mathrm{N}(\hat{\boldsymbol{\beta}}) &= \mathrm{S}(\hat{\boldsymbol{\beta}}) - \mathrm{C}(\hat{\boldsymbol{\beta}}).\end{aligned} \tag{72}$$

Since both $\frac{\partial \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})^\top}{\partial \hat{\boldsymbol{\beta}}}$ and $\mathrm{N}(\hat{\boldsymbol{\beta}})$ are unknown, the following update rule can be employed:

$$\hat{\boldsymbol{\beta}} - \check{\boldsymbol{\beta}} = -\left[\frac{\partial \boldsymbol{\theta}(\check{\boldsymbol{\beta}})^\top}{\partial \check{\boldsymbol{\beta}}}\mathrm{N}_\lambda(\check{\boldsymbol{\beta}})\frac{\partial \boldsymbol{\theta}(\check{\boldsymbol{\beta}})}{\partial \check{\boldsymbol{\beta}}^\top}\right]^{+} \frac{\partial \boldsymbol{\theta}(\check{\boldsymbol{\beta}})^\top}{\partial \check{\boldsymbol{\beta}}}\mathrm{N}_\lambda(\check{\boldsymbol{\beta}})\boldsymbol{\theta}(\check{\boldsymbol{\beta}}), \tag{73}$$

$$\mathrm{N}_\lambda(\check{\boldsymbol{\beta}}) = \mathrm{S}(\check{\boldsymbol{\beta}}) - \lambda_{min}\mathrm{C}(\check{\boldsymbol{\beta}}), \tag{74}$$

where λ_{min} is the smallest generalized eigenvalue solution of (41). Note that the Hessian matrix used in (74) is different than the one used in the least squares estimates.

Equation (73) has a different interpretation than (58). Suppose that we are given an estimate $\check{\boldsymbol{\beta}}$ and we are initially doing a minimization of the cost function over the parameter $\boldsymbol{\theta} \in \mathbb{R}^p$. Then, given $\check{\boldsymbol{\theta}} = \boldsymbol{\theta}(\check{\boldsymbol{\beta}})$, the updated estimator $\hat{\boldsymbol{\theta}}$ is obtained by solving the HEIV equation (41). To find the best approximation of $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})$, we minimize the cost function

$$\mathcal{J} = \frac{1}{2}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})\right)^\top \hat{\mathrm{C}}_{\hat{\theta}}^{+}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}(\hat{\boldsymbol{\beta}})\right), \tag{75}$$

estimating the covariance $\mathrm{C}_{\hat{\theta}}$ of the estimator $\hat{\boldsymbol{\theta}}$ similarly to (67).

3.5 The HEIV Estimator

The HEIV estimator is summarized next.

1. Compute an initial solution $\hat{\boldsymbol{\theta}}^{[0]}$, for example, the TLS estimate obtained assuming i.i.d. noise. Note

that a random initial value, however, suffices in most case to achieve convergence. If needed, find the corresponding $\hat{\boldsymbol{\beta}}^{[0]}$ by minimizing (75) and let $\hat{\hat{\boldsymbol{\theta}}}^{[0]} = \boldsymbol{\theta}(\hat{\boldsymbol{\beta}}^{[0]})$. Let $j = 0$.

2. Compute the current projection $\hat{\boldsymbol{z}}_i$ of $\boldsymbol{z}_i$ onto the manifold solution $\hat{\hat{\boldsymbol{\theta}}}^{[j]}$ and estimate the Lagrange multipliers $\boldsymbol{\eta}_i$ using (21).
3. Compute the weighted scatter matrix $\mathrm{S}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})$ using (36) and the weighted covariance matrix $\mathrm{C}(\hat{\hat{\boldsymbol{\theta}}}^{[j]})$ using (37).
4. Find the update estimate $\hat{\boldsymbol{\theta}}^{[j+1]}$ by solving the generalized eigenvalue problem (41) using generalized eigen(singular) value decomposition as discussed in Section 3.1. Update $j = j + 1$.
5. If it is necessary, estimate $\hat{\boldsymbol{\beta}}^{[j]}$ from $\hat{\boldsymbol{\theta}}^{[j]}$ by imposing (75) and recompute $\hat{\hat{\boldsymbol{\theta}}}^{[j]} = \boldsymbol{\theta}(\hat{\boldsymbol{\beta}}^{[j]})$.
6. Iterate through Steps 2 to 5 until $\lambda_{min} \approx 1$, up to a tolerance. Convergence is achieved usually in two or three iterations.

4 Relationship of the HEIV Equation to Other Techniques

The relationship of the HEIV algorithm with other estimators proposed in the literature for parameter estimation for constraints linear in the parameter $\boldsymbol{\theta}$ is established next. We prove that the HEIV estimator is the most general estimator which is optimal in the first order. In particular, the renormalization algorithm of Kanatani, the Sampson method, and the GTLS algorithm can be obtained from the general HEIV equation by making various simplifying assumptions.

4.1 GTLS Algorithm

Assume that the constraint is univariate and linear in both the measurements and parameter, i.e., $\boldsymbol{f}(\boldsymbol{z}_{io}, \boldsymbol{\theta}_o) = \boldsymbol{z}_{io}^\top \boldsymbol{\theta}_o$. Assume furthermore that the noise affecting the measurements is anisotropic, but homogeneous $\mathrm{C}_{z_i} = \mathrm{C}_z$. From $\Phi(\boldsymbol{z}_{io}) = \boldsymbol{z}_{io}^\top$ and $\boldsymbol{\varphi}(\boldsymbol{z}_{io}) = \boldsymbol{z}_{io}$, we have $\mathrm{J}_{f|\hat{z}_i}(\check{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = \hat{\boldsymbol{\theta}}$, thus

$$\mathrm{S}(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \frac{\boldsymbol{z}_i \boldsymbol{z}_i^\top}{\hat{\boldsymbol{\theta}}^\top \mathrm{C}_{z_i} \hat{\boldsymbol{\theta}}} = \frac{1}{\hat{\boldsymbol{\theta}}^\top \mathrm{C}_z \hat{\boldsymbol{\theta}}} \sum_{i=1}^{n} \boldsymbol{z}_i \boldsymbol{z}_i^\top. \tag{76}$$

Using $\mathrm{J}_{\phi|\hat{z}_i}(\check{\boldsymbol{z}}_i) = \mathrm{I}_p$,

$$\mathrm{C}(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \eta_i^2 \mathrm{C}_z = \left(\sum_{i=1}^{n} \eta_i^2 \right) \mathrm{C}_z. \tag{77}$$

Thus, the HEIV equation $\mathrm{S}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} = \mathrm{C}(\hat{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}$ can be written as

$$\begin{aligned} &\mathrm{S}_{GTLS}\hat{\boldsymbol{\theta}} = \lambda \mathrm{C}_{GTLS}\hat{\boldsymbol{\theta}} \quad \mathrm{S}_{GTLS} = \sum_{i=1}^{n} \boldsymbol{z}_i \boldsymbol{z}_i^\top \\ &\mathrm{C}_{GTLS} = \mathrm{C}_z \quad \lambda = (\hat{\boldsymbol{\theta}}^\top \mathrm{C}_z \hat{\boldsymbol{\theta}}) \sum_{i=1}^{n} \eta_i^2 \end{aligned} \tag{78}$$

and the solution $\hat{\boldsymbol{\theta}}$ sought is taken as the smallest generalized eigenvector of (78).

4.2 Sampson Method

In the Sampson method, we minimize the same objective function (12) when the constraint $\boldsymbol{f}$ is $\boldsymbol{f}(\boldsymbol{z}_{io}, \boldsymbol{\theta}_o) = \Phi(\boldsymbol{z}_{io})\boldsymbol{\theta}_o$. A key difference in the Sampson method is the assumption that the gradient of the cost function $\mathrm{J}_{\mathcal{J}|\hat{\theta}}(\hat{\boldsymbol{\theta}})$ can be computed by discarding the dependence of the scatter matrix $\mathrm{S}(\hat{\boldsymbol{\theta}})$ on the unknown value of $\hat{\boldsymbol{\theta}}$. Thus, by assuming $\mathrm{S}(\hat{\boldsymbol{\theta}}) \approx \mathrm{S}(\check{\boldsymbol{\theta}})$, where $\check{\boldsymbol{\theta}}$ is an already available estimate assumed constant, it is easy to see that the gradient becomes

$$\boldsymbol{J}_{\mathcal{J}_{SN}|\theta}(\hat{\boldsymbol{\theta}}) = \mathrm{S}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}, \tag{79}$$

thus the solution sought is obtained by the smallest eigenvector of

$$\mathrm{S}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}} = \lambda \hat{\boldsymbol{\theta}}. \tag{80}$$

Therefore, the Sampson method is obtained as a particular case of the HEIV equation obtained by substituting $\mathrm{C}(\check{\boldsymbol{\theta}}) = \mathrm{I}_p$. Discarding the dependency of the scatter matrix S in (79) on the *current* value of the parameter $\hat{\boldsymbol{\theta}}$ leads to a simplified calculation of the gradient, however, the solution obtained will be biased, even for $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_o$

$$E[\boldsymbol{J}_{\mathcal{J}_{SN}|\theta}(\boldsymbol{\theta}_o)] = E[\mathrm{S}(\boldsymbol{\theta}_o)]\, \boldsymbol{\theta}_o \neq \mathbf{0}. \tag{81}$$

We show next that any linearization of the constraint function $\Phi(\boldsymbol{z}_{io})\boldsymbol{\theta}_o = \mathbf{0}$ involving the parameter $\hat{\boldsymbol{\theta}}$ will result in a scheme equivalent with the Sampson method. Indeed, a first order approximation of the constraint $\boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}})$ around the current estimates $(\boldsymbol{z}_i, \check{\boldsymbol{\theta}})$ yields

$$\boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) = \boldsymbol{f}(\boldsymbol{z}_i, \check{\boldsymbol{\theta}}) + \mathrm{J}_{f|\hat{z}_i}(z_i, \check{\boldsymbol{\theta}})^\top (\hat{\boldsymbol{z}}_i - \boldsymbol{z}_i) + \Phi(\boldsymbol{z}_i)(\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}}). \tag{82}$$

From (12) and (82) and imposing $\boldsymbol{J}_{\mathcal{J}|\hat{\boldsymbol{z}}_i} = \mathbf{0}$, one obtains, after some manipulations,

$$\hat{\boldsymbol{z}}_i = \boldsymbol{z}_i - \mathrm{C}_{z_i} \mathrm{J}_{f|\hat{z}_i}(\boldsymbol{z}_i, \check{\boldsymbol{\theta}}) \mathrm{C}_f(\boldsymbol{z}_i, \check{\boldsymbol{\theta}})^+ \Phi(\boldsymbol{z}_i)\hat{\boldsymbol{\theta}}, \tag{83}$$

resulting in the cost function

$$\mathcal{J}_{SN}(\hat{\boldsymbol{\theta}}) = \frac{1}{2}\hat{\boldsymbol{\theta}}^\top \sum_{i=1}^{n} \Phi(\boldsymbol{z}_i)^\top \mathrm{C}_f(\boldsymbol{z}_i, \check{\boldsymbol{\theta}})^+ \Phi(\boldsymbol{z}_i)\hat{\boldsymbol{\theta}} = \frac{1}{2}\hat{\boldsymbol{\theta}}^\top \mathrm{S}(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}, \tag{84}$$

which has the gradient expressed in (79), resulting in the Sampson solution (80).

4.3 Renormalization Method

The *renormalization* method is an iterative method used to eliminate the bias present in the least squares solution by solving a generalized eigenvalue problem. The renormalization was proposed by Kanatani [20, pp. 267-295] and applied to a wide range of computer vision problems: conic fitting, optical flow, and rigid motion estimation. Kanatani [20] contains numerous applications of the renormalization together with a complete estimation framework.

We assume now that

$$\Phi(\boldsymbol{z}_i) = \mathrm{Z}_i \quad \mathrm{Z}_i = \begin{bmatrix} \boldsymbol{z}_i^{(1)\top} \\ \vdots \\ \boldsymbol{z}_i^{(m)\top} \end{bmatrix} \quad \boldsymbol{z}_i^\top = \left[\boldsymbol{z}_i^{(1)\top} \cdots \boldsymbol{z}_i^{(m)\top} \right] \tag{85}$$

and note that $\boldsymbol{\varphi}(\boldsymbol{z}_i) = \boldsymbol{z}_i$. Let $\sigma_\nu^2 \mathrm{C}_{z_i}^{(kl)} = \mathrm{cov}(\boldsymbol{z}_i^{(k)\top}, \boldsymbol{z}_i^{(l)\top})$ be the (kl)th block of the covariance $\sigma_\nu^2 \mathrm{C}_{z_i}$ of the measurements $\boldsymbol{z}_i$, where σ_ν^2 is a common factor, the noise variance, assumed unknown, which can be estimated after the main estimation process using (56). The covariance of the constraint (38) can be written as

$$\mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}}) \triangleq (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}})^\top \mathrm{C}_{z_i} (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}}) \tag{86}$$

function only on the covariance of the measurements and on the estimate $\hat{\boldsymbol{\theta}}$. The scatter (36) has the simpler expression

$$\mathrm{S}(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \mathrm{Z}_i^\top \mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}})^+ \mathrm{Z}_i. \tag{87}$$

The expected value $E[\mathrm{S}(\boldsymbol{\theta}_o)]\boldsymbol{\theta}_o$ can be expanded using (87) as

$$\begin{aligned} E[\mathrm{S}(\boldsymbol{\theta}_o)]\boldsymbol{\theta}_o &= E\left[\sum_{i=1}^{n} \mathrm{Z}_i^\top \mathrm{C}_{f_i}(\boldsymbol{\theta}_o)^+ \mathrm{Z}_i\right]\boldsymbol{\theta}_o \\ &= E\left[\sum_{i=1}^{n} (\mathrm{Z}_{io}^\top + \delta\mathrm{Z}_i^\top)\mathrm{C}_{f_i}(\boldsymbol{\theta}_o)^+ (\mathrm{Z}_{io} + \delta\mathrm{Z}_i)\right]\boldsymbol{\theta}_o, \end{aligned} \tag{88}$$

which can be written as

$$\begin{aligned} E[\mathrm{S}(\boldsymbol{\theta}_o)]\boldsymbol{\theta}_o &= \mathrm{S}_o(\boldsymbol{\theta}_o)\boldsymbol{\theta}_o + \sum_{i=1}^{n} \mathrm{C}_{f_i}(\boldsymbol{\theta}_o)^+ E[\delta\mathrm{Z}_i \delta\mathrm{Z}_i^\top]\boldsymbol{\theta}_o \\ &= \sigma_\nu^2 \sum_{i=1}^{n}\sum_{k,l=1}^{m} \gamma_i^{(kl)} \mathrm{C}_{z_i}^{(kl)} \boldsymbol{\theta}_o \neq \mathbf{0}, \end{aligned} \tag{89}$$

where $\gamma_i^{(kl)}$ is the (kl)th element of the matrix $\mathrm{C}_{f_i}(\boldsymbol{\theta})^+$. In the renormalization method, the bias of the gradient of the objective function is removed by defining the unbiased scatter matrix

$$\begin{aligned} \mathrm{N}_K(\boldsymbol{\theta}_o) &= \mathrm{S}(\boldsymbol{\theta}_o) - \sigma_\nu^2 \sum_{i=1}^{n}\sum_{k,l=1}^{m} \gamma_i^{(kl)} \mathrm{C}_{z_i}^{(kl)} \\ &= \sum_{i=1}^{n}\sum_{k,l=1}^{m} \gamma_i^{(kl)} \left(\boldsymbol{z}_i^{(k)} \boldsymbol{z}_i^{(l)\top} - \sigma_\nu^2 \mathrm{C}_{z_i}^{(kl)}\right). \end{aligned} \tag{90}$$

Let

$$\mathrm{C}_K(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n}\sum_{k,l=1}^{m} \gamma_i^{(kl)} \mathrm{C}_{z_i}^{(kl)} \quad \mathrm{N}_K(\hat{\boldsymbol{\theta}}) = \mathrm{S}(\hat{\boldsymbol{\theta}}) - \sigma_\nu^2 \mathrm{C}_K(\hat{\boldsymbol{\theta}}). \tag{91}$$

An unbiased least squares estimation can be obtained when the objective function (84) is replaced by the objective function

$$\mathcal{J}_K(\hat{\boldsymbol{\theta}}) = \frac{1}{2}\hat{\boldsymbol{\theta}}^\top \mathrm{N}_K(\check{\boldsymbol{\theta}})\hat{\boldsymbol{\theta}}, \tag{92}$$

which is minimized under the constraint $\|\hat{\boldsymbol{\theta}}\| = 1$. The renormalization seeks a solution by iteratively solving the following eigenvalue equation

$$\mathrm{N}_K(\check{\boldsymbol{\theta}})\, \hat{\boldsymbol{\theta}} = \lambda \hat{\boldsymbol{\theta}}. \tag{93}$$

We prove next that the renormalization equation approximates the HEIV equation on average. Since the scatter matrix expression $\mathrm{S}(\hat{\boldsymbol{\theta}})$ is identical in both HEIV and the renormalization, we will show that $\mathrm{C}(\hat{\boldsymbol{\theta}})$ is proportional with $\mathrm{C}_K(\hat{\boldsymbol{\theta}})$. Indeed, from (21), we have

$$\boldsymbol{\eta}_i \boldsymbol{\eta}_i^\top = \mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}})^+ (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}})^\top \boldsymbol{z}_i \boldsymbol{z}_i^\top (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}}) \mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}})^+$$

having the expected value

$$E[\boldsymbol{\eta}_i \boldsymbol{\eta}_i^\top] = \mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}})^+ (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}})^\top E[\boldsymbol{z}_i \boldsymbol{z}_i^\top] (\mathrm{I}_m \otimes \hat{\boldsymbol{\theta}}) \mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}})^+ = \sigma_\nu^2 \mathrm{C}_{f_i}(\hat{\boldsymbol{\theta}})^+. \tag{94}$$

We have assumed unbiased parameter estimates, zero-mean noise, and have discarded moments higher than two. Thus, $E[\eta_{ik}\eta_{il}] = \sigma_\nu^2 \gamma_i^{(kl)}$. The coefficients multiplying $\mathrm{C}_i^{(kl)}$ in the renormalization process are proportional to the expected values of the coefficients used in the HEIV equation, the proportionality factor being given by the equivalent noise variance σ_ν^2.

4.4 The Fundamental Numerical Scheme

Chojnacki et al. proposed [5], [6], [7] a different approach called *Fundamental Numerical Scheme* (FNS) as an alternative for solving (11). The FNS was derived only for scalar constraints, $m = 1$. Assuming that, in the objective function from (12), we add the Lagrange multiplier corresponding to the constraint $\|\hat{\boldsymbol{\theta}}\| = 1$

$$\begin{aligned} \mathcal{J}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_n) = &\frac{1}{2}\sum_{i=1}^{n} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i)^\top \mathrm{C}_{z_i}^{+} (\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i) \\ &+ \sum_{i=1}^{n} \boldsymbol{\eta}_i^\top \boldsymbol{f}(\hat{\boldsymbol{z}}_i, \hat{\boldsymbol{\theta}}) - \lambda \hat{\boldsymbol{\theta}}^\top \hat{\boldsymbol{\theta}}, \end{aligned}$$

we similarly obtain from the derivation in Section 2 that the gradient of the objective function is

$$\mathrm{J}_{\mathcal{J}|\hat{\theta}}(\hat{\boldsymbol{\theta}}) = [\mathrm{S}(\hat{\boldsymbol{\theta}}) - \mathrm{C}(\hat{\boldsymbol{\theta}})]\, \hat{\boldsymbol{\theta}} - \lambda \hat{\boldsymbol{\theta}}.$$

Hence, in order to obtain a minimum of the objective function at $\hat{\boldsymbol{\theta}}$, we have to find the solution to the equation

$$\left[\mathrm{S}(\check{\boldsymbol{\theta}}) - \mathrm{C}(\check{\boldsymbol{\theta}})\right]\, \hat{\boldsymbol{\theta}} = \lambda\, \hat{\boldsymbol{\theta}}. \tag{95}$$

The refined solution $\hat{\boldsymbol{\theta}}$ corresponds to the smallest in absolute value eigenvalue of the matrix $\mathrm{S}(\check{\boldsymbol{\theta}}) - \mathrm{C}(\check{\boldsymbol{\theta}})$, which most closely approximates the null space of $\mathrm{S}(\check{\boldsymbol{\theta}}) - \mathrm{C}(\check{\boldsymbol{\theta}})$. It is assumed that, with each step of (95), a better estimate of this null space is obtained. When convergence is reached, the smallest eigenvalue of (95) must necessarily be zero. The initial solution employed in the FNS is the total least squares solution.

A comparison between the FNS and HEIV scheme reveals the close relationship between the two estimates. The methods employ different numerical solutions (ordinary eigenproblem versus generalized singular value decomposition) to find a solution to the same cost function. Experimental results showed that FNS and HEIV, when initialized from the same solution, yield very close numerical results [7], with the HEIV estimate being slightly better. It was also noted that the HEIV scheme is more robust to the initial solution (convergence to the same solution is obtained by starting with random parameter) than the FNS. On the other hand, the FNS scheme it is not affected by the singularity posed by the perfect measurements as the HEIV scheme is.

4.5 Levenberg-Marquardt Method

A widely employed optimization technique for least squares problems is the Levenberg-Marquardt (LM) method [34, pp. 683-689]. An excellent review of the topic is done by Triggs et al. in [38]. Assume that the ideal values of the measurements $\boldsymbol{y}_{io} \in \mathbb{R}^q$, $\boldsymbol{x}_{io} \in \mathbb{R}^m$, $i = 1, \ldots, n$ obey the constraint

$$\boldsymbol{y}_{io} = \boldsymbol{g}(\boldsymbol{x}_{io}, \boldsymbol{\beta}_o) \qquad i = 1, \ldots, n \tag{96}$$

and that *only* $\boldsymbol{y}_{io}$ was corrupted by noise

$$\boldsymbol{y}_i = \boldsymbol{y}_{io} + \delta\boldsymbol{y}_i \quad \delta\boldsymbol{y}_i \sim GI(\mathbf{0}, \sigma_\nu^2 \boldsymbol{C}_{y_i}).$$

In the least squares method, the estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}_o$ is found by minimizing the sum of the squared Mahalanobis distances

$$\mathcal{J}(\hat{\boldsymbol{\theta}}, = \frac{1}{2}\sum_{i=1}^{n}\Big(\boldsymbol{y}_i - \boldsymbol{g}(\boldsymbol{x}_{io}, \hat{\boldsymbol{\beta}})\Big)^\top \mathrm{C}_{y_i}^{+}\Big(\boldsymbol{y}_i - \boldsymbol{g}(\boldsymbol{x}_{io}, \hat{\boldsymbol{\beta}})\Big). \tag{97}$$

Define

$$\delta\boldsymbol{Y}(\hat{\boldsymbol{\beta}}) \triangleq \begin{bmatrix} \boldsymbol{y}_1 - \boldsymbol{g}(\boldsymbol{x}_{1o}, \hat{\boldsymbol{\beta}}) \\ \vdots \\ \boldsymbol{y}_n - \boldsymbol{g}(\boldsymbol{x}_{no}, \hat{\boldsymbol{\beta}}) \end{bmatrix} \quad \delta\boldsymbol{Y}(\hat{\boldsymbol{\beta}}) \in \mathbb{R}^{mn},$$
$$\mathrm{C}_y = \mathbf{bdiag}(\mathrm{C}_{y_1}, \ldots, \mathrm{C}_{y_n}) \quad \mathrm{C}_y \in \mathbb{R}^{mn \times mn}.$$

The gradient vector is

$$\begin{aligned} \frac{\partial \mathcal{J}(\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}} &= \mathrm{J}_{\mathcal{J}|\hat{\beta}} \mathrm{C}_y^{+} \delta\boldsymbol{Y}(\hat{\boldsymbol{\beta}}) \\ \mathrm{J}_{\mathcal{J}|\hat{\beta}} &= -\left[\frac{\partial \boldsymbol{g}(\boldsymbol{x}_{1o}, \hat{\boldsymbol{\beta}})^\top}{\partial \hat{\boldsymbol{\beta}}}, \ldots, \frac{\partial \boldsymbol{g}(\boldsymbol{x}_{no}, \hat{\boldsymbol{\beta}})^\top}{\partial \hat{\boldsymbol{\beta}}}\right] \in \mathbb{R}^{p \times mn}. \end{aligned} \tag{98}$$

The Hessian matrix becomes

$$\begin{aligned} \frac{\partial^2 \mathcal{J}(\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}} \partial \hat{\boldsymbol{\beta}}^\top} &= \mathrm{J}_{\mathcal{J}|\hat{\beta}} \mathrm{C}_y^{+} \mathrm{J}_{\mathcal{J}|\hat{\beta}}^\top \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{\partial^2 g_i(\boldsymbol{x}_{1o}, \hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}}\, \partial \hat{\boldsymbol{\beta}}^\top} \Big[\mathrm{C}_{y_i}^{+}\Big(\boldsymbol{y}_i - \boldsymbol{g}(\boldsymbol{x}_{io}, \hat{\boldsymbol{\beta}})\Big)\Big]_j, \end{aligned} \tag{99}$$

where $g_i(\cdot)$ is the ith component of $g(\cdot)$ and $[x]_j$ is the jth component of the vector $\boldsymbol{x}$. In practice, the approximation

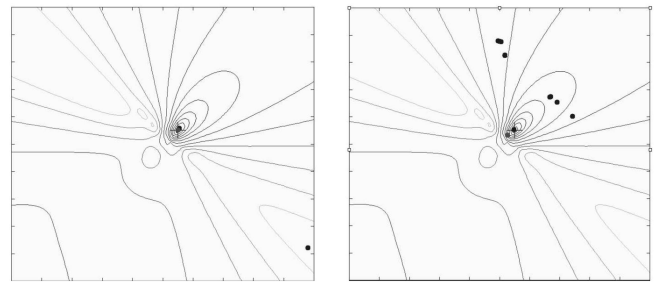
$$\frac{\partial^2 \mathcal{J}(\hat{\boldsymbol{\beta}})}{\partial \hat{\boldsymbol{\beta}} \partial \hat{\boldsymbol{\beta}}^\top} \approx \mathrm{J}_{\mathcal{J}|\hat{\boldsymbol{\beta}}} \mathrm{C}_y^{+} \mathrm{J}_{\mathcal{J}|\hat{\boldsymbol{\beta}}}^\top$$

is used. This approximation is valid for small errors $\delta\boldsymbol{y}_i$ in the metric defined by C_{y_i} or for functions $\boldsymbol{g}(\boldsymbol{x}_{io}, \hat{\boldsymbol{\beta}})$ which are quasi-linear in the parameter $\hat{\boldsymbol{\beta}}$. During the LM iterations, the update $\delta\hat{\boldsymbol{\beta}}$ is computed by solving the following linear system:

$$\Big(\mathrm{J}_{\mathcal{J}|\hat{\beta}} \mathrm{C}_y^{+} \mathrm{J}_{\mathcal{J}|\hat{\beta}}^\top + \kappa \mathrm{W}\Big)\delta\hat{\boldsymbol{\beta}} = -\mathrm{J}_{\mathcal{J}|\hat{\beta}} \mathrm{C}_y^{+} \delta\boldsymbol{Y}(\hat{\boldsymbol{\beta}}), \tag{100}$$

where W is chosen most often to be the identity matrix I_p and κ is a constant which is used to switch between a Gauss-Newton step, obtained for small κ and a gradient descent step when $\kappa \gg 0$ depending on the error surface encountered. The exact path followed in the error surface to reach the final solution depends on the initial value $\hat{\boldsymbol{\beta}}^{[0]}$ and on the heuristics employed to modify the dampening factor κ.

The LM algorithm requires a good initialization to converge to the correct solution. On the other hand, the HEIV is less dependent on the initialization and converges in fewer steps to the final solution. In order to assess the relative performance of the LM and HEIV, we have performed a synthetic experiment for a linear problem $y_{io} = \mathrm{x}_{io}^\top \boldsymbol{\xi}_o + \alpha_o = 0$, $\boldsymbol{\beta}_o = [\boldsymbol{\xi}_o, \alpha_o]^\top \in \mathbb{R}^3$ in which the ideal values $\boldsymbol{z}_{io} = [\boldsymbol{x}_{io}, y_{io}]^\top$, satisfying $\boldsymbol{z}_{io}^\top \boldsymbol{\beta}_o = 0$, were affected by heteroscedastic noise



(a) (b)

Fig. 3. Comparison between the convergence of the (a) HEIV algorithm and (b) LM algorithm. Dots represent iterations steps on the error surface. Note the more sinuous path of the LM when initialized from further away from the ideal solution denoted by a cross.

$$\delta\boldsymbol{z}_i \sim GI\Big(\mathbf{0}, 0.2^2 \mathrm{R}_i^\top \mathrm{diag}\Big([z_{io}]_1^2, [z_{io}]_2^2, [z_{io}]_3^2\Big)\mathrm{R}_i\Big),$$

where R_i is a rotation matrix randomly chosen. The experiments employed 30 measurements randomly selected for $\boldsymbol{\beta}_o = [1\,1\,1]^\top$ with $[\boldsymbol{x}_{io}]_{1,2} \in [-10, 10]$. For LM, we had employed the implementation provided by the optimization toolbox in Matlab and, for HEIV, we had employed our own implementation in Matlab. In Fig. 3, the convergence steps (denoted by dots) for the HEIV and LM in a typical trial show the path toward convergence. While the LM requires, in general, about 20-30 iterations, the HEIV converges in two to three iterations.

5 Applications

In this section, we describe applications of the HEIV estimator and compare it with estimators such as total least squares (TLS), Levenberg-Marquardt solutions, etc. The HEIV estimator is obtained from the linearized algorithm followed by the application of additional constraints if it is necessary. In each case, the constraint can be written as separable in the parameter $\hat{\boldsymbol{\theta}}$.

5.1 Camera Calibration

Camera calibration was discussed in Section 1. The solution has to satisfy (7)

$$\mathrm{A}(\hat{\boldsymbol{z}}_i, \boldsymbol{m}_{io})\hat{\boldsymbol{p}} = \Phi(\hat{\boldsymbol{z}}_i)\hat{\boldsymbol{\theta}} = \mathbf{0} \quad \|\hat{\boldsymbol{\theta}}\| = 1 \quad i = 1, \ldots, n, \tag{101}$$

where $\Phi(\hat{\boldsymbol{z}}_i)$ is a 2×12 matrix and $\hat{\boldsymbol{\theta}}$ is a 12-dimensional vector with $\hat{\boldsymbol{\theta}}$ having norm one to remove the scale ambiguity. The projection matrix $\hat{\mathrm{P}}$ is computed by reshuffling the vector $\hat{\boldsymbol{\theta}}$ and the intrinsic and extrinsic camera parameters $\hat{\mathrm{K}}$, respectively, $\hat{\mathrm{R}}$, $\hat{\boldsymbol{t}}$ are obtained from the QR decomposition of $\hat{\mathrm{P}}$

$$\hat{\mathrm{P}} = \hat{\mathrm{K}}\big[\hat{\mathrm{R}} \quad \hat{\boldsymbol{t}}\big], \tag{102}$$

with $\hat{\mathrm{K}}$ being an upper triangular matrix and $\hat{\mathrm{R}}$, $\hat{\boldsymbol{t}}$ the rotation and translation of the camera with respect to the world coordinates. Further additional constraints on $\hat{\mathrm{K}}$, like known aspect ratio of the pixels, can also be imposed. In general, it can be assumed that $\hat{k}_{12} = 0$.

The calibration error is defined as the norm of the 10-dimensional error vector of the four intrinsic and six extrinsic parameters. For each measure, 500 trials of 50 points

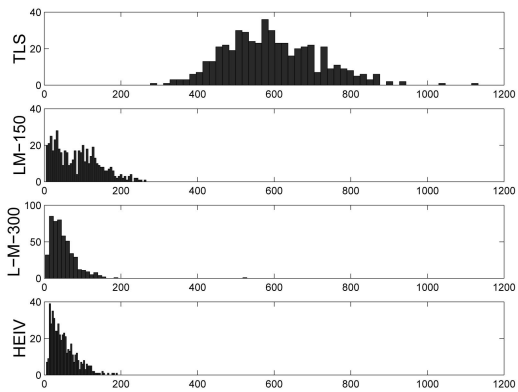


Fig. 4. Histogram of calibration error norms. $\sigma_\nu = 0.5$.

were performed with the ideal camera parameters being $f = 117$, $k_x = k_y = 1$, $O_x = 250$, and $O_y = 250$.

The TLS solution is biased because each row in (9) has a different covariance matrix. In Fig. 4, the increased error norm is clearly shown in the TLS case. When the data is normalized, as is recommended for TLS [16, p. 170], its performance improves significantly but remains slightly biased.

The Levenberg-Marquardt algorithm starts with the total least squares solution and, subsequently, refines it by minimizing the following cost function:

$$\mathcal{J}(\hat{\mathrm{P}}) = \sum_{i=1}^{n} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2 \text{ subject to } \hat{z}_{1i} = \frac{\boldsymbol{q}_1^\top \boldsymbol{M}_{io}}{\boldsymbol{q}_3^\top \boldsymbol{M}_{io}} \;\; \hat{z}_{2i} = \frac{\boldsymbol{q}_2^\top \boldsymbol{M}_{io}}{\boldsymbol{q}_3^\top \boldsymbol{M}_{io}}, \tag{103}$$

where $\boldsymbol{q_k}^\top$ is the kth row of the matrix P. Because the image points have i.i.d. covariance matrices, the optimization function is without weights. The output of the LM algorithm is shown after 150 and 300 function evaluations in Fig. 4. The LM algorithm requires about 300 function evaluations to converge. The HEIV algorithm starts with a random initial solution and converges in about three iterations. The LM and HEIV estimates require about the same amount of computation.

5.2 Three-Dimensional Rigid Motion

Estimating the rigid motion transformation between two 3D point clouds is a fundamental problem in 3D scene reconstruction from unregistered 3D data acquired by range sensors or stereo heads. Two popular closed-form estimators are based on the SVD method proposed originally by Arun et al. [1] and refined later by Umeyama [40], respectively, using the quaternion method proposed by Horn et al. [17]. Both estimators were shown to numerically yield the same results [10], [30] since they utilize different representation for the same constraint solved by the SVD and the eigenvalue decomposition. The SVD and quaternion-based representation are optimal only when the 3D measurements are affected by i.i.d. noise. When this assumption is violated, both algorithms yield estimates having a significant bias even when a large number of points are available. In practice, the recovered 3D measurements are affected by strong heteroscedastic noise. For example, for range data, the 3D data is acquired by emitting a radiation pulse (e.g., laser) toward the target and measuring the time elapsed between sending and receiving the pulse. Thus, the noise affecting the data is mainly along the line of sight between the sensor and the scene. Also, when stereo heads are employed, the triangulated 3D points are affected also by errors along the viewing direction with noise increasing with the distance between the points and the stereo head.

Let the two sets of ideal, noise-free matched 3D measurements be $\mathrm{U}_o = \{\boldsymbol{u}_{1o}, \boldsymbol{u}_{2o}, \ldots, \boldsymbol{u}_{no}\}$ and $\mathrm{V}_o = \{\boldsymbol{v}_{1o}, \boldsymbol{v}_{2o}, \ldots, \boldsymbol{v}_{no}\}$. The available 3D measurements $\boldsymbol{u}_i$, $\boldsymbol{v}_i$ are

$$\begin{aligned} &\boldsymbol{u}_i = \boldsymbol{u}_{io} + \delta\boldsymbol{u}_i \quad \boldsymbol{v}_i = \boldsymbol{v}_{io} + \delta\boldsymbol{v}_i \\ &\delta\boldsymbol{u}_i \sim GI(\boldsymbol{0}, \sigma_\nu^2 \mathrm{C}_{u_i}) \quad \delta\boldsymbol{v}_i \sim GI(\boldsymbol{0}, \sigma_\nu^2 \mathrm{C}_{v_i}). \end{aligned} \tag{104}$$

The ideal values must obey the rigid motion constraint

$$\boldsymbol{v}_{io} = \mathrm{R}_o \boldsymbol{u}_{io} + \boldsymbol{t}_o, \tag{105}$$

where R_o is the 3×3 rotation matrix and $\boldsymbol{t}_o$ is the translation vector. The estimate $\hat{\mathrm{R}}$ of R_o must satisfy the constraint $\hat{\mathrm{R}}\hat{\mathrm{R}}^\top = \mathrm{I}_3$.

An elegant rotation parametrization is obtained by using quaternions, which are four-dimensional unit vectors $\boldsymbol{q} = [\, q_0 \;\; q_1 \;\; q_2 \;\; q_3 \,]^\top$. It can be shown [33] that the constraint (105) is equivalent to

$$\mathrm{Z}_{io}\boldsymbol{q}_o + \boldsymbol{\alpha}_o = \boldsymbol{0}, \tag{106}$$

where $\mathrm{Z}_{io} \in \mathbb{R}^{3\times 4}$ is the matrix of the carriers

$$\mathrm{Z}_{io} = \begin{bmatrix} v_{1io} - u_{1io} & 0 & -v_{3io} - u_{3io} & v_{2io} + u_{2io} \\ v_{2io} - u_{2io} & v_{3io} + u_{3io} & 0 & -v_{1io} - u_{1io} \\ v_{3io} - u_{3io} & -v_{2io} - u_{2io} & v_{1io} + u_{1io} & 0 \end{bmatrix} \tag{107}$$

and the intercept

$$\boldsymbol{\alpha}_o = \mathrm{Q}_o \boldsymbol{t}_o \qquad \mathrm{Q}_o = \begin{bmatrix} -q_{0o} & -q_{3o} & q_{2o} \\ q_{3o} & -q_{0o} & -q_{1o} \\ -q_{2o} & q_{1o} & -q_{0o} \end{bmatrix}. \tag{108}$$

The matrix Q_o is nonsingular as long as q_{0o} is different from zero since the determinant of Q_o is equal to $-q_{0o}$. Because $q_{0o} = 0$ represents a discontinuity in the quaternion representation, the application of this approach is not recommended for cases when the rotation angle is close to $\pm\pi$ [27]. Note that (106) is of the form (4) with $\mathrm{z}_i^\top = [\boldsymbol{u}_i^\top \; \boldsymbol{v}_i^\top]$ and $\boldsymbol{\theta}_o^\top = [\boldsymbol{q}_o^\top \; \boldsymbol{\alpha}_o^\top]$ and that $\Phi(\boldsymbol{z}_{io})$ is linear in the measurements $\boldsymbol{z}_i$.

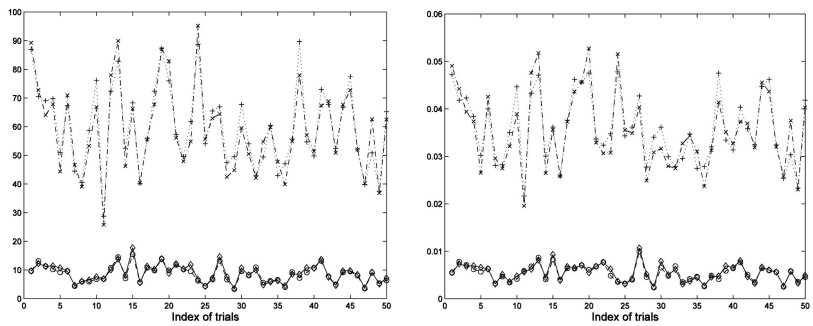
The quaternion method is a TLS estimator in which the quaternion $\hat{\boldsymbol{q}}$ is obtained as the smallest eigenvector of the matrix

$$\begin{aligned} &\mathrm{M} = \sum_{i=1}^{n} \tilde{\mathrm{Z}}_i^\top \tilde{\mathrm{Z}}_i \\ &\tilde{\mathrm{Z}}_i = \begin{bmatrix} \tilde{v}_{1i} - \tilde{u}_{1i} & 0 & -\tilde{v}_{3i} - \tilde{u}_{3i} & \tilde{v}_{2i} + \tilde{u}_{2i} \\ \tilde{v}_{2i} - \tilde{u}_{2i} & \tilde{v}_{3i} + \tilde{u}_{3i} & 0 & -\tilde{v}_{1i} - \tilde{u}_{1i} \\ \tilde{v}_{3i} - \tilde{u}_{3i} & -\tilde{v}_{2i} - \tilde{u}_{2i} & \tilde{v}_{1i} + \tilde{u}_{1i} & 0 \end{bmatrix}, \end{aligned}$$

where

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i - \tilde{\boldsymbol{u}} \quad \tilde{\boldsymbol{u}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{u}_i \quad \tilde{\boldsymbol{v}}_i = \boldsymbol{v}_i - \tilde{\boldsymbol{v}} \quad \tilde{\boldsymbol{v}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_i \tag{109}$$

are the centered measurements. The rotation estimate $\hat{\mathrm{R}}$ is obtained uniquely from $\hat{\boldsymbol{q}}$, $\|\hat{\boldsymbol{q}}\| = 1$, using the equation,



(a) (b)

Fig. 5. Comparison between the bootstrap (BT) and Monte Carlo (MC) error estimates for HEIV and the quaternion method. (a) Translation. (b) Rotation. "$\diamond$" MC estimate for HEIV, "$\circ$" BT estimate for HEIV, "$+$" MC estimate for quaternion, and "x" BT estimate for quaternion.

$$\mathrm{R} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_2q_1+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_3q_1-q_0q_2) & 2(q_3q_2+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}. \tag{110}$$

The translation estimate is $\hat{\boldsymbol{t}} = \tilde{\boldsymbol{v}} - \hat{\mathrm{R}}\tilde{\boldsymbol{u}}$. By exploiting the relationship between (106) and (4), the HEIV estimator can be applied directly.

5.2.1 Experiments with Synthetic Data

The simulated setting used in our experiments consists of a stereo head moved around a fixed scene. The cameras had zero vergence and focal distance $f = 536$, yielding a field of view of 50° on both x and y axes. The baseline of the stereo head was 100 and the image planes were 500×500, all values being in pixel units. The $n = 50$ three-dimensional points were uniformly generated inside a cube with the side length 800 placed at 1,300 in front of the cameras. The 3D points are projected onto the image planes, corrupted by adding normal noise with $\sigma_\nu^2 = 1$, and then allocated to the nearest lattice site. The 3D information is recovered using Kanatani's triangulation method [20, pp. 171-186]. For this type of triangulation, there are closed, form expressions for C_{v_i} and C_{u_i}. Alternatively, the *bootstrap* method described in [27] can be used.

The performance evaluation of the quaternion and HEIV-based estimators was done using Monte Carlo trials and the bootstrap using sampling of residuals $\boldsymbol{z}_i - \tilde{\boldsymbol{z}}_i$, whitened by the corresponding covariance of the residuals. Note that the Monte Carlo analysis assumes that the ideal values of the estimates (R_o and $\boldsymbol{t}_o$) are known, while the bootstrap does not. We had employed $B = 200$ samples for both Monte Carlo and bootstrap methods. For each Monte Carlo trial b, the estimates are denoted by $\hat{\mathrm{R}}^b, \hat{\boldsymbol{t}}^b$, while, for the bootstrap, the estimates are denoted by $\hat{\mathrm{R}}^{*b}, \hat{\boldsymbol{t}}^{*b}$.

Fifty trials, each having the motion parameters randomly generated (chosen such that the scene remains in the field of view of the cameras) and different 3D point configurations were performed. For each trial, the translation error ϵ_t^2 is estimated as the average of $\|\hat{\boldsymbol{t}}^b - \boldsymbol{t}_o\|$ for the Monte Carlo and the average of $\|\hat{\boldsymbol{t}}^{*b} - \hat{\boldsymbol{t}}\|$, $b = 1, \ldots, B$ for the bootstrap. The rotation forms a multiplicative group, thus, the rotation estimation error ϵ_R^2 is defined as the angle-axis representation of the rotation matrix $\hat{\mathrm{R}}\mathrm{R}_o^\top$. In the *angle-axis representation*, using the notations from [19, pp. 100-103], R is represented by the vector $\boldsymbol{r} = \Omega\, \boldsymbol{l}$, $\boldsymbol{r} \in \mathbb{R}^3$, where

$$\Omega = \|\boldsymbol{r}\| = \arccos\left[\frac{\mathrm{trace}(\mathrm{R}) - 1}{2}\right] \quad \text{and}$$
$$\boldsymbol{l} = \frac{\boldsymbol{l}'}{\|\boldsymbol{l}'\|} \quad \boldsymbol{l}' = \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}. \tag{111}$$

We denote this operation by $\boldsymbol{r} \triangleq \mathbf{ax}(\mathrm{R})$. It follows that the Monte Carlo rotation error is the average of $\|\mathbf{ax}(\hat{\mathrm{R}}^{\mathrm{b}}\mathrm{R}_o^\top)\|$, while the bootstrap rotation error is the average of $\|\mathbf{ax}(\hat{\mathrm{R}}^{*\mathrm{b}}\hat{\mathrm{R}}^\top)\|$.

In Fig. 5, the translation and rotation error for each 50 trials defined as above are plotted for the HEIV and quaternion algorithms using the bootstrap and the Monte Carlo estimates. Note the excellent agreement between the bootstrap and Monte Carlo error estimates and the larger translation and rotation error yielded by the quaternion method compared with the HEIV algorithm.

5.2.2 Experiments with Real Data

To evaluate the HEIV base's rigid motion estimator, the quaternion method, and renormalization, we have used two sequences for which ground truth information was available, the *Castle* sequence from the *CIL-CMU* database which has a significant translation but a relatively small rotation, and the *PUMA* sequence, which has a dominant rotation component. Typical frames for the Castle sequence and the PUMA sequence are shown in Fig. 6.

From each experiment we have performed, four frames were selected. Since the motion between the frames is known, two "virtual" stereo-heads can be defined using the first two and the last two frames. For each pair, 3D measurements are obtained using triangulation and the uncertainty of the triangulated points extracted using bootstrap. The relative pose between the locations of the two stereo-heads is subsequently estimated from the two sets of 3D points. The information flow is summarized in Fig. 7, with a specific motion estimator applied to the 3D data to estimate the rotation and translation between the two stereo-heads.

For the HEIV, quaternion, and renormalization methods, the bootstrapped covariance matrices of $\hat{\boldsymbol{t}}$ and $\hat{\mathrm{R}}$ are used to define the average estimation error as

$$\hat{\epsilon}_t^2 = \mathrm{trace}(\hat{\mathrm{C}}_t) \quad \hat{\epsilon}_R^2 = \mathrm{trace}(\hat{\mathrm{C}}_R). \tag{112}$$

The average estimation error for the rotation and translation using the frames 5, 7, 9, and 11 from the *Castle*

(a) (b) (c) (d)

Fig. 6. Typical frames from the sequences with the matched points labeled. (a) and (b) *Castle* frames from the CIL-CMU database. (c) and (d) Frames from the *PUMA* sequence.

sequence are presented in Table 1. There were 93 points matched across the four frames using [44].

The estimation errors for the HEIV are much smaller than for the quaternion method which is heavily biased under heteroscedastic noise. The HEIV errors are also smaller than the renormalization method, which failed to converge in few trials. The bootstrapped covariance matrices of the estimates $\hat{\mathrm{R}}$, $\hat{\boldsymbol{t}}$ for the quaternion and HEIV methods are plotted in Fig. 8. As expected, the quaternion-based algorithm has a much larger variability than the HEIV since it assumes i.i.d. data.

For the *PUMA* sequence, the frames 9, 10, 12, and 13 were used and 79 matched points retained. The results are shown in Table 2 and illustrate the superior performance of the HEIV algorithm. The renormalization method failed to converge to the correct solution.

6 Discussion

We next discuss several applications of the HEIV algorithm proposed in the literature. The origin of the algorithm, derived for univariate constraints, can be traced to Leedan and Meer [23]. In [23], a slightly simplified version of the approach described was applied to fundamental matrix estimation and to conic fitting. The authors reported improved behavior of the algorithm compared to other known techniques.

The HEIV estimator, in a form similar to the one described in this paper, was presented originally in [28] and applied to camera calibration and fundamental matrix estimation. The authors reported improved behavior of the HEIV-based estimator compared with techniques based on the Levenberg-Marquardt algorithm initialized by the linearized solution obtained using the total least squares estimator.

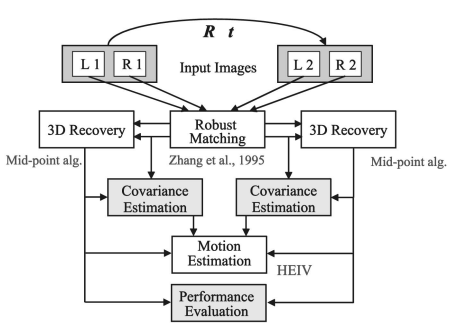


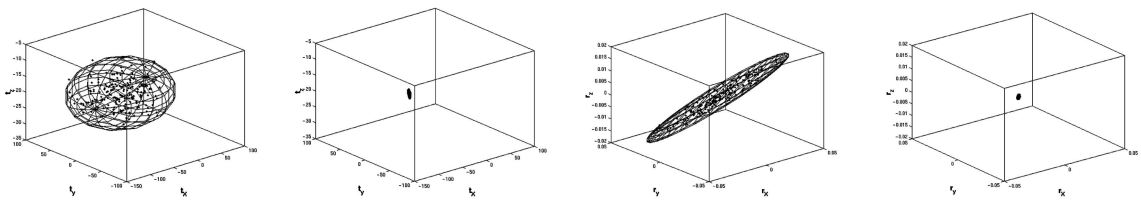
Fig. 7. Information flow in the experimental setup for estimating rigid motion using real data obtained from a stereo head. Covariance of the recovered 3D measurements is obtained using error propagation or bootstrap.

It is important to point out that the HEIV provides a different optimization approach for handling nonlinear constraints of a particular type $\phi(\boldsymbol{x})^\top \boldsymbol{\theta}_o = \mathbf{0}$ relating the parameter $\boldsymbol{\theta}_o$ to the measurements compared to the usage of generic tools such as the *fmincon* function from the optimization toolbox within MATLAB. This translates into a particular expression of the gradient, which can be solved through the eigenvalue or singular value decomposition. The nature of the the constraint and the manner in which the optimization is carried out result in a large numerical robustness of the HEIV algorithm to the quality of the initial solution. The usage of *fmincon* function involves implicitly extending the parameter space to include the nuisance parameters, similar to the Gold Standard Method proposed by Hartley. Moreover, in order to achieve convergence to the correct solution, generally, a good initialization would be required. Rather than increasing the dimensionality of the problem, the HEIV relies on factoring out the nuisance parameters and solving for the parameter only. Thus, the resulted dimensionality of the space over which the HEIV seeks the parameter estimate is constant and independent on the number of measurements. Employing out of the box optimization packages, such as *fmincon* without addressing the sparseness of the Hessian matrix induced by the nuisance parameters would result in inefficient and slow execution time.

In [29], the HEIV algorithm was applied to conic fitting. Simulations with synthetic and real data showed significantly improved performance compared with the direct least squares (DLS) method and was more numerically robust compared with the minimization of geometric

TABLE 1
Performance Evaluation: *Castle* Sequence

Algorithm	$\hat{\epsilon}_t$	$\hat{\epsilon}_R$
HEIV	1.82	0.010
Renormalization	8.42	0.024
Quaternion	70.31	0.038



(a) (b) (c) (d)

Fig. 8. Uncertainty of the translation and rotation estimates for the quaternion and HEIV methods evaluated by bootstrap. (a) and (c) Translation and rotation for the quaternion method. (b) and (d) Translation and rotation for the HEIV-based estimator. The ellipsoids assure a 0.95 coverage for the ideal rotation and translation.

distances with the LM algorithm initialized from the DLS solution. Confidence regions derived from the estimated covariance matrices of the HEIV estimator were showed to be in close correspondence with the bootstrapped confidence regions using resampling of residuals.

The HEIV algorithm was applied for trifocal tensor estimation in [26]. The authors demonstrated the robustness of the HEIV algorithm for different sequences with respect to the choice of the initial solution. For hard cases, the HEIV was shown to be superior compared with the Gold-Standard approach which failed to converge to the correct solution in five out of 500 trials due to the bad initialization provided by the linearized solution yielded by the TLS algorithm. For stable configurations of the scene and camera, the HEIV solution and the Gold Standard method were shown to be numerically very close, with the HEIV providing a faster convergence.

In [3], the HEIV estimator was applied to 2D rigid motion between two images. The approach was further extended by Nestares and Fleet for affine motion estimation [32]. The authors reported a significant improvement of the accuracy of the optical flow estimated with the HEIV scheme due to the non i.i.d. nature of the noise affecting the measurements.

Förstner published an optimal algorithm for 3D points or 3D lines recovery from 2D points or 2D lines [11]. The recovery of lines was achieved with Plücker coordinates. The author followed the HEIV algorithm presented by Matei and Meer in [28], however, he obtained an ordinary eigenvalue problem instead of a generalized one. Uncertainty of the estimated quantities was also obtained in terms of covariance matrices.

The HEIV algorithm was used by Georgescu and Meer to initialize the bundle adjustment of parameters for the recovery of 3D structure and camera motion from uncalibrated video sequences [13]. Experiments using simulated and real sequences showed the improved behavior of the final estimates when the HEIV was employed at various stages, such as trifocal tensor estimation, camera resectioning, in terms of faster convergence of the bundle adjustment when initialized with the HEIV estimates, and no instances of failed convergence to the correct solution.

TABLE 2
Performance Evaluation: *PUMA* Sequence

Algorithm	$\hat{\epsilon}_t$	$\hat{\epsilon}_R$
HEIV	0.82	1.153
Renormalization	20.1	2.71
Quaternion	29.6	2.16

Subbarao et al. [37] employed the HEIV algorithm for estimating the camera pose for 3D tracking within an augmented reality application. The authors reported superior accuracy compared to using the Levenberg-Marquardt optimization.

We encourage the readers to apply the HEIV algorithm for all the tasks in which linearized algorithms are typically employed, since the final bundle adjustment of parameters can benefit from a more accurate and numerically robust initial solution. Implementations of the HEIV algorithm in C++ and Matlab are freely available to download at http://www.caip.rutgers.edu/riul/research/hetero.html.

APPENDIX

The Kronecker product and the *vec* operator provide an elegant mathematical framework for vector calculus and are extensively used throughout the paper. The survey [2] or the book [15] contain all the necessary background and also cover vector calculus.

Given a matrix $\mathrm{A} \in \mathbb{R}^{p\times q}$, the vectorization of A is defined as the pq-dimensional vector

$$\mathrm{A} = [\,\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \cdots \quad \boldsymbol{a}_q\,] \quad \boldsymbol{a}_i \in \mathbb{R}^p \quad \mathrm{vec}(\mathrm{A}) = \begin{bmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_q \end{bmatrix}. \quad (\mathrm{A}.1)$$

The Kronecker product between the matrices A and $\mathrm{B} \in \mathbb{R}^{s\times t}$ is defined as the $ps \times qt$ matrix

$$\mathrm{A} \otimes \mathrm{B} = \begin{bmatrix} A_{11}\mathrm{B} & \cdots & A_{1q}\mathrm{B} \\ \cdots & A_{ij}\mathrm{B} & \cdots \\ A_{p1}\mathrm{B} & \cdots & A_{pq}\mathrm{B} \end{bmatrix}. \quad (\mathrm{A}.2)$$

The ijth block $A_{ij}\mathrm{B}$ is called the ij partition of the matrix $\mathrm{A} \otimes \mathrm{B}$.

Let the following elements have the dimensions

$$\begin{array}{ccccc} \mathrm{A}, \mathrm{H} \in \mathbb{R}^{p\times q} & \mathrm{B}, \mathrm{R} \in \mathbb{R}^{s\times t} & \mathrm{C} \in \mathbb{R}^{r\times l} & \mathrm{D} \in \mathbb{R}^{q\times s} & \mathrm{F} \in \mathbb{R}^{q\times u} \\ \mathrm{G} \in \mathbb{R}^{t\times u} & \mathrm{M} \in \mathbb{R}^{m\times m} & \mathrm{N} \in \mathbb{R}^{n\times n} & \mathrm{Q} \in \mathbb{R}^{s\times p} & \boldsymbol{z} \in \mathbb{R}^{s} \end{array}$$

"$\mathrm{trace}(\cdot)$" stand for the trace, "$\det(\cdot)$" for the determinant, "$\mathrm{rank}(\cdot)$" for the rank of a matrix, and $\mathrm{I}_p \in \mathbb{R}^{p\times p}$ for the identity matrix.

1.1 Properties of the Kronecker Product

$$
\begin{aligned}
(\mathrm{A}\otimes\mathrm{B})\otimes\mathrm{C} &= \mathrm{A}\otimes(\mathrm{B}\otimes\mathrm{C}), && \text{(A.3)}\\
(\mathrm{A}\otimes\mathrm{B})^\top &= \mathrm{A}^\top\otimes\mathrm{B}^\top, && \text{(A.4)}\\
(\mathrm{A}+\mathrm{H})\otimes(\mathrm{B}+\mathrm{R}) &= \mathrm{A}\otimes\mathrm{B}+\mathrm{A}\otimes\mathrm{R}\\
&\quad +\mathrm{H}\otimes\mathrm{B}+\mathrm{H}\otimes\mathrm{R}, && \text{(A.5)}\\
(\mathrm{A}\otimes\mathrm{B})(\mathrm{D}\otimes\mathrm{G}) &= \mathrm{AD}\otimes\mathrm{BG}, && \text{(A.6)}\\
\mathrm{trace}(\mathrm{N}\otimes\mathrm{M}) &= \mathrm{trace}(\mathrm{N})\mathrm{trace}(\mathrm{M}), && \text{(A.7)}\\
\det(\mathrm{N}\otimes\mathrm{M}) &= [\det(\mathrm{N})]^m[\det(\mathrm{M})]^n, && \text{(A.8)}\\
\mathrm{rank}(\mathrm{A}\otimes\mathrm{B}) &= \mathrm{rank}(\mathrm{A})\mathrm{rank}(\mathrm{B}), && \text{(A.9)}\\
(\mathrm{I}_s\otimes\boldsymbol{z})\mathrm{B} &= \mathrm{B}\otimes\boldsymbol{z}. && \text{(A.10)}
\end{aligned}
$$

1.2 Properties of the *vec* Operator

$$
\begin{aligned}
\mathrm{vec}(\boldsymbol{z}) &= \mathrm{vec}(\boldsymbol{z}^\top), && \text{(A.11)}\\
\mathrm{vec}(\mathrm{A}+\mathrm{H}) &= \mathrm{vec}(\mathrm{A})+\mathrm{vec}(\mathrm{H}), && \text{(A.12)}\\
\mathrm{vec}(\mathrm{AD}) &= (\mathrm{I}_s\otimes\mathrm{A})\mathrm{vec}(\mathrm{D}) = (\mathrm{D}^\top\otimes\mathrm{I}_p)\mathrm{vec}(\mathrm{A})\\
&= (\mathrm{D}^\top\otimes\mathrm{A})\mathrm{vec}(\mathrm{I}_q), && \text{(A.13)}\\
\mathrm{trace}(\mathrm{A}^\top\mathrm{H}) &= [\mathrm{vec}(\mathrm{A})]^\top\mathrm{vec}(\mathrm{H}). && \text{(A.14)}
\end{aligned}
$$

1.3 Vector Calculus

The Jacobian of a vector valued function $\boldsymbol{f}(\boldsymbol{z}) \in \mathbb{R}^m$ in the variable $\boldsymbol{z} \in \mathbb{R}^s$ is the $s \times m$ matrix

$$
\mathrm{J}_{f|z} \triangleq \frac{\partial \boldsymbol{f}(\boldsymbol{z})^\top}{\partial \boldsymbol{z}} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \cdots & \frac{\partial f_m}{\partial z_1} \\ \vdots & & \vdots \\ \frac{\partial f_1}{\partial z_s} & \cdots & \frac{\partial f_m}{\partial z_s} \end{bmatrix} = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{z})}{\partial \boldsymbol{z}^\top}\right)^\top. \quad \text{(A.15)}
$$

For a scalar valued function $f(\boldsymbol{z})$, the Jacobian becomes $\mathrm{J}_{f|z} = \nabla f$, the gradient with respect to $\boldsymbol{z}$. The Jacobian of the composite function $\boldsymbol{f}(\boldsymbol{z}(\boldsymbol{y}))$, $\boldsymbol{y} \in \mathbb{R}^q$, is computed using the chain rule

$$
\mathrm{J}_{f|y} = \mathrm{J}_{z|y}\mathrm{J}_{f|z}. \quad \text{(A.16)}
$$

Let $\boldsymbol{g}(\boldsymbol{z}) \in \mathbb{R}^m$ be another vector valued function, then

$$
h(\boldsymbol{z}) = \boldsymbol{f}(\boldsymbol{z})^\top\boldsymbol{g}(\boldsymbol{z}) \quad \mathrm{J}_{h|z} = \nabla h = \mathrm{J}_{g|z}\boldsymbol{f}(\boldsymbol{z}) + \mathrm{J}_{f|z}\boldsymbol{g}(\boldsymbol{z}). \quad \text{(A.17)}
$$

Define the rows of the matrix $\mathrm{A}(\boldsymbol{z})$ as $\mathrm{A}^\top = [\boldsymbol{a}'_1(\boldsymbol{z}) \cdots \boldsymbol{a}'_p(\boldsymbol{z})]$, $\boldsymbol{a}'_j(\boldsymbol{z}) \in \mathbb{R}^q$, and the q-dimensional vector $\boldsymbol{b}(\boldsymbol{z})$, then

$$
\boldsymbol{f}(\boldsymbol{z}) = \mathrm{A}(\boldsymbol{z})\boldsymbol{b}(\boldsymbol{z}) \quad \mathrm{J}_{f|z} = \left[\mathrm{J}_{a'_1|z}\boldsymbol{b}(\boldsymbol{z}) \cdots \mathrm{J}_{a'_p|z}\boldsymbol{b}(\boldsymbol{z})\right] + \mathrm{J}_{b|z}\mathrm{A}(\boldsymbol{z})^\top. \quad \text{(A.18)}
$$

The particular cases

$$
\begin{aligned}
&\boldsymbol{f}(\boldsymbol{z}) = \mathrm{A}\boldsymbol{z} \quad \mathrm{J}_{f|z} = \mathrm{A}^\top \text{ and } f(\boldsymbol{z}) = \boldsymbol{z}^\top\mathrm{A}\boldsymbol{z}\\
&\quad \mathrm{J}_{f|z} = \nabla f = (\mathrm{A}+\mathrm{A}^\top)\boldsymbol{z}
\end{aligned} \quad \text{(A.19)}
$$

are frequently used throughout the paper.

We can show that $\boldsymbol{\theta}$ and $\mathrm{J}_{\mathcal{J}|\theta}(\boldsymbol{\theta})$ are orthogonal, i.e.,

$$
\boldsymbol{\theta}^\top\mathrm{J}_{\mathcal{J}|\theta}(\boldsymbol{\theta}) = 0 \quad \forall\boldsymbol{\theta}. \quad \text{(B.1)}
$$

Indeed, using one of the Moore-Penrose conditions of a pseudoinverse [14, p. 257],

$$
\begin{aligned}
\boldsymbol{\theta}^\top\,\mathrm{C}(\boldsymbol{\theta})\,\boldsymbol{\theta} &= \sum_{i=1}^n \boldsymbol{\theta}^\top(\boldsymbol{\eta}_i\otimes\mathrm{I}_p)^\top\mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)(\boldsymbol{\eta}_i\otimes\mathrm{I}_p)\boldsymbol{\theta}\\
&= \sum_{i=1}^n \boldsymbol{\eta}_i^\top(\mathrm{I}_m\otimes\boldsymbol{\theta})^\top\mathrm{C}_\varphi(\check{\boldsymbol{z}}_i)(\mathrm{I}_m\otimes\boldsymbol{\theta})\boldsymbol{\eta}_i\\
&= \sum_{i=1}^n \boldsymbol{\eta}_i^\top\mathrm{C}_f(\check{\boldsymbol{z}}_i,\boldsymbol{\theta})\boldsymbol{\eta}_i\\
&= \sum_{i=1}^n \boldsymbol{f}(\boldsymbol{z}_i,\boldsymbol{\theta})^\top\mathrm{C}_f(\check{\boldsymbol{z}}_i,\boldsymbol{\theta})^+\mathrm{C}_f(\check{\boldsymbol{z}}_i,\boldsymbol{\theta})\mathrm{C}_f(\check{\boldsymbol{z}}_i,\boldsymbol{\theta})^+\boldsymbol{f}(\boldsymbol{z}_i,\boldsymbol{\theta})\\
&= \boldsymbol{\theta}^\top\,\mathrm{S}(\boldsymbol{\theta})\,\boldsymbol{\theta}.
\end{aligned}
$$

Acknowledgments

The support of the US National Science Foundation under the grants IRI 95-30546 and IRI 99-87695 is gratefully acknowledged. The authors thank Bogdan Georgescu for fruitful discussions and for the C++ implementation of the algorithm. This work was done while Dr. Matei was with Rutgers University.

References

[1] K. Arun, T. Huang, and S. Blostein, "Least-Squares Fitting of Two 3D Point Sets," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 5, pp. 698-700, Sept. 1987.

[2] J.W. Brewer, "Kronecker Products and Matrix Calculus in System Theory," *IEEE Trans. Circuits and Systems,* vol. 25, no. 9, pp. 772-781, June 1978.

[3] J. Bride and P. Meer, "Registration via Direct Methods: A Statistical Approach," *Proc. 2001 IEEE Conf. Computer Vision and Pattern Recognition,* pp. 984-989, Dec. 2001.

[4] H. Chen and P. Meer, "Robust Regression with Projection Based M-Estimators," *Proc. Ninth Int'l Conf. Computer Vision,* vol. 2, pp. 878-885, Oct. 2003.

[5] W. Chojnacki, M.J. Brooks, and A. van den Hengel, "Rationalising the Renormalisation Method of Kanatani," *J. Math. Imaging and Vision,* vol. 14, pp. 21-38, Feb. 2001.

[6] W. Chojnacki, M.J. Brooks, A. van den Hengel, and D. Gawley, "On the Fitting of Surfaces to Data with Covariances," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 11, pp. 1294-1303, Nov. 2000.

[7] W. Chojnacki, M.J. Brooks, A. van den Hengel, and D. Gawley, "From FNS to HEIV: A Link between Two Vision Parameter Estimation Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 264-268, Feb. 2004.

[8] W. Chojnacki, M.J. Brooks, A. van den Hengel, and D. Gawley, "A New Constrained Parameter Estimator for Computer Vision Applications," *Image and Vision Computing,* vol. 22, pp. 85-91, 2004.

[9] A. Edelman, T.A. Arias, and S.T. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM J. Matrix Analysis Application,* vol. 20, pp. 217-234, 1997.

[10] D. Eggert, A. Lorusso, and R. Fisher, "Estimating 3D Rigid Body Transformations: A Comparison of Four Major Algorithms," *Machine Vision and Applications,* vol. 9, pp. 272-290, June 1997.

[11] W. Förstner, "On Estimating 3D Points and Lines from 2D Points and Lines," *Festschrift Anläßlich des 60. Geburtstages von Prof. Dr. Ing. Bernhard Wrobel,* Technische Univ. Darmstadt, pp. 69-87, 2001.

[12] W. Fuller, *Measurement Error Models.* John Wiley and Sons, 1987.

[13] B. Georgescu and P. Meer, "Balanced Recovery of 3D Structure and Camera Motion from Uncalibrated Image Sequences," *Proc. European Conf. Computer Vision,* vol. II, pp. 294-308, 2002.

[14] G. Golub and C.F. Van Loan, *Matrix Computations,* third ed. The Johns Hopkins Univ. Press, 1996.

[15] A. Graham, *Kronecker Products and Matrix Calculus: With Applications.* Ellis Horwood series in mathematics and its applications, 1981.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge Univ. Press, 2000.
[17] B. Horn, H. Hilden, and S. Negahdaripour, "Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices," *J. Optical Soc. of Am.,* vol. 5, pp. 1127-1135, 1988.
[18] S.V. Huffel and J. Vandewalle, "Analysis and Properties of GTLS in Problem $AX \approx B$ When Some or All Columns of A Are Subject to Errors," *SIAM J. Matrix Analysis and Applications,* vol. 10, pp. 294-315, July 1989.
[19] K. Kanatani, *Geometric Computation for Machine Vision.* Oxford Science Publications, 1993.
[20] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice.* Elsevier, 1996.
[21] K. Kanatani and D. Morris, "Gauges and Gauge Transformations for Uncertainty Description of Geometric Structure with Indeterminacy," *IEEE Trans. Information Theory,* vol. 47, pp. 2017-2028, 2001.
[22] J. Koenderink and A. van Doorn, "The Generic Bilinear Calibration-Estimation Problem," *Int'l J. Computer Vision,* vol. 23, no. 2, pp. 303-353, 1998.
[23] Y. Leedan and P. Meer, "Heteroscedastic Regression in Computer Vision: Problems with Bilinear Constraint," *Int'l J. Computer Vision,* vol. 37, no. 2, pp. 127-150, June 2000.
[24] Y. Ma, J. Kosecka, and S. Sastry, "Optimization Criteria and Geometric Algorithms for Motion and Structure Estimation," *Int'l J. Computer Vision,* vol. 44, pp. 219-249, 2001.
[25] B. Matei, "Heteroscedastic Errors-in-Variables Models in Computer Vision," PhD thesis, Dept. of Electrical and Computer Eng., Rutgers Univ., 2001, http://www.caip.rutgers.edu/riul/research/theses.html.
[26] B. Matei, B. Georgescu, and P. Meer, "A Versatile Method for Trifocal Tensor Estimation," *Proc. Eighth Int'l Conf. Computer Vision,* vol. II, pp. 578-585, July 2001.
[27] B. Matei and P. Meer, "Optimal Rigid Motion Estimation and Performance Evaluation with Bootstrap," *Proc. Computer Vision and Pattern Recognition Conf.,* pp. 339-345, June 1999.
[28] B. Matei and P. Meer, "A General Method for Errors-in-Variables Problems in Computer Vision," *Proc. Computer Vision and Pattern Recognition Conf.,* vol. II, pp. 18-25, June 2000.
[29] B. Matei and P. Meer, "Reduction of Bias in Maximum Likelihood Ellipse Fitting," *Proc. 15th Int'l Conf. Computer Vision and Pattern Recognition,* vol. III, pp. 802-806, Sept. 2000.
[30] B. Matei, P. Meer, and D. Tyler, "Performance Assessment by Resampling: Rigid Motion Estimators," *Empirical Evaluation Techniques in Computer Vision,* pp. 72-95, 1998.
[31] D. Morris, K. Kanatani, and T. Kanade, "Gauge Fixing for Accurate 3D Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. II, pp. 343-350, 2001.
[32] O. Nestares and D. Fleet, "Errors-in-Variables Likelihood Functions for Motion Estimation," *Proc. IEEE Int'l Conf. Image Processing,* 2003.
[33] N. Ohta and K. Kanatani, "Optimal Estimation of Three-Dimensional Rotation and Reliability Evaluation," *Proc. European Conf. Computer Vision (ECCV '98),* pp. 175-187, June 1998.
[34] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C,* second ed. Cambridge Univ. Press, 1992.
[35] P. Sampson, "Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm," *Computer Graphics and Image Processing,* vol. 18, no. 1, pp. 97-108, Jan. 1982.
[36] G.W. Stewart, "Errors in Variables for Numerical Analysis," *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling,* pp. 3-10, 1997.
[37] R. Subbarao, P. Meer, and Y. Genc, "A Balanced Approach to 3D Tracking from Image Streams," *Proc. IEEE and ACM Int'l Symp. Mixed and Augmented Reality,* pp. 70-78, Oct. 2005.
[38] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon, "Bundle Adjustment—A Modern Synthesis," *Vision Algorithms: Theory and Practice,* pp. 298-372, 2000.
[39] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, 1998.
[40] S. Umeyama, "Least-Squares Estimation of Transformation Parameters between Two Point Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 4, pp. 376-380, Apr. 1991.
[41] A. van den Hengel, W. Chonacki, M.J. Brooks, and D. Gawley, "A New Constrained Parameter Estimator: Experiments in Fundamental Matrix Computation," *Proc. British Machine Vision Conf.,* pp. 468-476, Sept. 2002.
[42] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem. Computational Aspects and Analysis.* Soc. for Industrial and Applied Math., 1991.
[43] R.H. Zamar, "Robust Estimation in the Errors-in-Variables Model," *Biometrika,* vol. 76, pp. 149-160, 1989.
[44] Z. Zhang, "Image Matching Software," 1995, http://www.inria.fr/robotvis/personnel/zzhang.

Bogdan C. Matei received the DiplEngn and MSc degrees in electrical and computer engineering from the Polytechnic University of Bucharest, Romania, in 1994 and 1995, respectively, and the PhD in electrical engineering from Rutgers University in 2001. Between 1994 and 1997, he was with the Department of Electronics and Telecommunications at the Polytechnic University of Bucharest as a teaching fellow and research associate. In 1994, he was awarded a TEMPUS scholarship funded by the European Union at Polytechnics of Turin, Italy, to research optical character recognition using neural networks. He received research fellowships at the Technical University of Darmstadt, Germany, in 1995 and 1996. From 1997 until 2001, he was with the Center for Advanced Information Processing affiliated with Rutgers University, researching the application of modern statistical methods for optimal parameter estimation under heteroscedastic noise and performance evaluation of computer vision algorithms. From 2001 until the present, he has been with the Vision Technologies Group from Sarnoff Corporation, first as a member of the technical staff and lately as a senior member of the technical staff. His research interests include object recognition, statistical learning, video-based aerial and ground autonomous navigation, real-time aerial video surveillance, and georegistration. He has coauthored more than 15 papers and book chapters in the areas of computer vision and holds several patents awarded or pending. He was in the program committee at numerous international conferences and workshops such as CVPR, ICCV, and ECCV. He received the best student paper award at the IEEE Computer Vision and Pattern Recognition Conference in 1999. He is a member of the IEEE Computer Society.

Peter Meer received the DiplEngn degree from the Bucharest Polytechnic Institute, Romania, in 1971 and the DSc degree from the Technion, Israel Institute of Technology, Haifa, in 1986, both in electrical engineering. From 1971 to 1979, he was with the Computer Research Institute, Cluj, Romania, working on R&D of digital hardware. From 1986 and 1990, he was an assistant research scientist at the Center for Automation Research, University of Maryland at College Park. In 1991, he joined the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, New Jersey, and is currently a professor. He has held visiting appointments in Japan, Korea, Sweden, Israel, and France, and was on the organizing committees of numerous international workshops and conferences. He was an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* between 1998 and 2002, a member of the editorial board of *Pattern Recognition* between 1990 and 2005, and was a guest editor of *Computer Vision and Image Understanding* for a special issue on robust statistical techniques in image understanding in 2000. He is the coauthor of an award winning paper in pattern recognition in 1989, the best student paper in the 1999, and the best paper in the 2000 IEEE Conference on Computer Vision and Pattern Recognition. His research interest is in the application of modern statistical methods to image understanding problems. He is a senior member of the IEEE and the IEEE Computer Society.