Dimensionality reduction for more stable vision parameter estimation

T. Scoleri[1,2]   W. Chojnacki[2]   M.J. Brooks[2]

[1]Defence Science and Technology Organisation, Edinburgh, SA 5111, Australia
[2]School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia
E-mail: tony.scoleri@dsto.defence.gov.au

Abstract: The problem of estimating parameters from data is considered for a class of multi-objective models of importance in computer vision. One previous approach for solving the problem is via the fundamental numerical scheme (FNS). Here, a more stable version of FNS is developed, with better convergence properties than the original version. The improvement in performance is achieved by reducing the original estimation problem to a couple of problems of lower dimension. By way of example, the new algorithm is applied to the problem of estimating the trifocal tensor relating three views of a scene. Experiments carried out with both synthetic and real images reveal the new estimator to be more stable compared to the original FNS method, and commensurate in accuracy with, but faster than, the gold standard maximum likelihood estimator.

1 Introduction

Fitting parametric models to data is a ubiquitous task in computer vision. In a typical model, parameters describe a relationship among image feature locations. The parameters and image data pertaining to the model are combined in a system of equations

$$f(x, \theta) = 0 \qquad (1)$$

where x is a length-k vector describing an ideal data point, θ is a length-l vector of parameters, and $f(x, \theta)$ is a length-m vector of multi-objective constraints of the form

$$f(x, \theta) = U(x)^{\mathrm{T}} \theta \qquad (2)$$

where $U(x)$ is an $l \times m$ data carrier matrix with entries formed by polynomial functions in $[x^{\mathrm{T}}, 1]^{\mathrm{T}}$. Models with a multi-component constraint vector of this type include ones in which parameters describe such entities as a planar homography [1], trifocal and quadrifocal tensors [2–4], a camera projection matrix [5], and an aggregate of affine subspaces [6, 7]. If observed data points $x_1, \ldots, x_n$ come equipped with covariance matrices $\Lambda_{x_1}, \ldots, \Lambda_{x_n}$ quantifying measurement errors in the data, then a statistically meaningful estimate of θ based on the compound set of the data points and their covariances can be obtained by minimising the multi-objective approximated maximum likelihood (AML) cost function

$$J_{\mathrm{AML}}(\theta; x_1, \ldots, x_n) = \sum_{i=1}^{n} f(x_i, \theta)^{\mathrm{T}} \Sigma(x_i, \theta)^{-1} f(x_i, \theta)$$

where $\Sigma(x_i, \theta) = \partial_x f(x_i, \theta) \Lambda_{x_i} [\partial_x f(x_i, \theta)]^{\mathrm{T}}$. Importantly, when the length m of the $f(x_i, \theta)$ surpasses the common codimension r of the submanifolds of the form $\{x \in \mathbb{R}^k | f(x, \theta) = 0\}$ with θ representing ideal parameters that might have generated the data, the inverses $\Sigma(x_i, \theta)^{-1}$ in the above expression must be replaced, say, by the r-truncated pseudo-inverses $\Sigma(x_i, \theta)_r^+$ to obtain a meaningful cost function [8, 9]. The AML estimate of θ, defined as the minimiser of the function $\theta \mapsto J_{\mathrm{AML}}(\theta; x_1, \ldots, x_n)$ and denoted $\hat{\theta}_{\mathrm{AML}}^u$, exhibits nearly optimal statistical behaviour and − unlike the maximum likelihood estimate − is relatively inexpensive to compute [10]. As a consequence of $\theta \mapsto J_{\mathrm{AML}}(\theta; x_1, \ldots, x_n)$ being homogeneous of degree zero, $\hat{\theta}_{\mathrm{AML}}^u$ is determined only up to scale. Once $\hat{\theta}_{\mathrm{AML}}^u$ has been generated, additional constraints − if they apply − involving the parameters alone can be accommodated via an adjustment procedure. In what follows we shall confine our attention to the estimation phase that precedes adjustment, concentrating effectively on unconstrained minimisation of J_{AML} (this is

218

IET Comput. Vis., 2008, Vol. 2, No. 4, pp. 218–227

underlined by attaching the superscript u in the symbol for the AML estimate).

Various methods are available for finding $\hat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$. One is the fundamental numerical scheme (FNS) introduced in [11] for single-objective models and extended in [10] to the case of multi-objective models. Another is the heteroscedastic errors-in-variables (HEIV) scheme initially proposed by Leedan and Meer [12] for single-equation models and further developed by Matei and Meer [5] to cover multi-equation models. Both techniques estimate the parameters of the model iteratively. To ensure convergence, the methods often require a good initial parameter estimate, but sometimes even an accurate seed leads to divergence if the level of noise in the data is too high.

The main purpose of this paper is to present a reduced form of FNS, where only a subset of the total parameter vector is estimated iteratively and the remaining parameters are recovered in a single step based on the result of the earlier iterative process. The reduced algorithm in effect replaces the original estimation problem with a couple of problems of lower dimension. The algorithm is an extension to the multi-objective setting of the reduced single-objective FNS given in [13]. As is demonstrated in the experimental section of the paper, the process of dimension reduction leads to significant benefits. Compared with the full form [10], the reduced form of the algorithm requires a less accurate initial estimate and enjoys better convergence properties. Although the paper is primarily concerned with FNS, the optimality condition which underlies the reduced form of this algorithm can readily be exploited to advance a reduced form of HEIV. In the experimental section, results for both the reduced FNS and reduced HEIV are presented.

2 Variational equation

The minimiser $\hat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ satisfies the necessary optimality condition

$$[\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}}(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^u_{\mathrm{AML}}} = \mathbf{0}^{\mathrm{T}} \qquad (3)$$

with $\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}}$ the row vector of the partial derivatives of J_{AML} with respect to $\boldsymbol{\theta}$. This is termed the variational equation. With the aid of (2) reformulated as

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) = (\boldsymbol{\theta}^{\mathrm{T}} \otimes \boldsymbol{I}_m)\mathrm{vec}(\boldsymbol{U}^{\mathrm{T}}) \qquad (4)$$

where $\boldsymbol{I}_m$ denotes the $m \times m$ identity matrix, and vec and $\otimes$ denote the vectorisation and Kronecker product operators [14], respectively, it can be shown that

$$[\partial_{\boldsymbol{\theta}} J_{\mathrm{AML}}(\boldsymbol{\theta}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)]^{\mathrm{T}} = 2X_{\boldsymbol{\theta}} \boldsymbol{\theta}$$

where $X_{\boldsymbol{\theta}} = M_{\boldsymbol{\theta}} - N_{\boldsymbol{\theta}}$ is an $l \times l$ symmetric matrix with

$$M_{\boldsymbol{\theta}} = \sum_{i=1}^{n} U_i \Sigma_i^{-1} U_i^{\mathrm{T}} \qquad (5a)$$

$$N_{\boldsymbol{\theta}} = \sum_{i=1}^{n} (\boldsymbol{I}_l \otimes \boldsymbol{\eta}_i^{\mathrm{T}}) B_i (\boldsymbol{I}_l \otimes \boldsymbol{\eta}_i) \qquad (5b)$$

$$U_i = U(\boldsymbol{x}_i)$$

$$B_i = \partial_{\boldsymbol{x}_i} \mathrm{vec}(U_i^{\mathrm{T}}) \Lambda_{\boldsymbol{x}_i} [\partial_{\boldsymbol{x}_i} \mathrm{vec}(U_i^{\mathrm{T}})]^{\mathrm{T}} \qquad (5c)$$

$$\Sigma_i = (\boldsymbol{\theta}^{\mathrm{T}} \otimes \boldsymbol{I}_m) B_i (\boldsymbol{\theta} \otimes \boldsymbol{I}_m) \qquad (5d)$$

$$\boldsymbol{\eta}_i = \Sigma_i^{-1} U_i^{\mathrm{T}} \boldsymbol{\theta} \qquad (5e)$$

The variational equation (3) can accordingly be rewritten as

$$X_{\boldsymbol{\theta}} \boldsymbol{\theta} = \mathbf{0} \qquad (6)$$

where the evaluation at $\hat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$ is dropped for clarity. In this form, the variational equation can readily serve as a basis for isolating $\hat{\boldsymbol{\theta}}^u_{\mathrm{AML}}$, as will be expanded upon below.

3 Reduced variational equation

Assume that the carrier matrix $U(\boldsymbol{x})$ can be written as

$$U(\boldsymbol{x}) = \begin{bmatrix} Z(\boldsymbol{x}) \\ W \end{bmatrix} \qquad (7)$$

where $Z(\boldsymbol{x})$ is an $(l - m) \times m$ matrix that depends on $\boldsymbol{x}$ (a 'pure measurement' matrix), and W is an $m \times m$ invertible matrix that does not depend on $\boldsymbol{x}$. Corresponding to this splitting of $U(\boldsymbol{x})$, the parameter vector $\boldsymbol{\theta}$ will be subdivided as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\alpha} \end{bmatrix} \qquad (8)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ are vectors of length $l - m$ and m, respectively. The partitioning of $U(\boldsymbol{x})$ and $\boldsymbol{\theta}$ reflects the fact that some components of $\boldsymbol{\theta}$, considered as indeterminates, appear in each of the equations of (1) only with constant coefficients. The vector $\boldsymbol{\alpha}$ collects together those components of $\boldsymbol{\theta}$ that appear in (1) with pure constant coefficients. For each $i = 1, \ldots, m$, the non-zero entries of the ith column of W represent the constant coefficients of the components of $\boldsymbol{\alpha}$ in the ith equation of (1). If, for instance, every equation of (1) has exactly one parameter with a unity coefficient, then, after re-ordering of the equations of (1) if necessary, it can be assumed that $W = \boldsymbol{I}_m$.

We shall now present a system of two equations that jointly are equivalent to the variational equation (6). One of these equations involves only $\boldsymbol{\mu}$ and can be solved separately, and the other expresses $\boldsymbol{\alpha}$ in terms of $\boldsymbol{\mu}$. We begin by noting

that, in view of (7)

$$\mathrm{vec}(\boldsymbol{U}(\boldsymbol{x})^{\mathrm{T}}) = \begin{bmatrix} \mathrm{vec}(\boldsymbol{Z}(\boldsymbol{x})^{\mathrm{T}}) \\ \mathrm{vec}(\boldsymbol{W}^{\mathrm{T}}) \end{bmatrix}$$

so that

$$\partial_{\boldsymbol{x}}\mathrm{vec}(\boldsymbol{U}(\boldsymbol{x})^{\mathrm{T}}) = \begin{bmatrix} \partial_{\boldsymbol{x}}\mathrm{vec}(\boldsymbol{Z}(\boldsymbol{x})^{\mathrm{T}}) \\ \boldsymbol{0}_{m^2 \times k} \end{bmatrix}$$

Hence, for each $i = 1, \ldots, n$, the $lm \times lm$ matrix $\boldsymbol{B}_i$ can be represented as

$$\boldsymbol{B}_i = \begin{bmatrix} \boldsymbol{B}_i^0 & \boldsymbol{0}_{(l-m)m \times m^2} \\ \boldsymbol{0}_{m^2 \times (l-m)m} & \boldsymbol{0}_{m^2 \times m^2} \end{bmatrix} \qquad (9)$$

where $\boldsymbol{B}_i^0$ is the $(l-m)m \times (l-m)m$ matrix given by

$$\boldsymbol{B}_i^0 = \partial_{\boldsymbol{x}_i}\mathrm{vec}(\boldsymbol{Z}_i^{\mathrm{T}})\Lambda_{\boldsymbol{x}_i}[\partial_{\boldsymbol{x}_i}\mathrm{vec}(\boldsymbol{Z}_i^{\mathrm{T}})]^{\mathrm{T}}, \quad \boldsymbol{Z}_i = \boldsymbol{Z}(\boldsymbol{x}_i)$$

Define an $m \times m$ matrix Σ_i' by

$$\Sigma_i' = (\boldsymbol{\mu}^{\mathrm{T}} \otimes \boldsymbol{I}_m)\boldsymbol{B}_i^0(\boldsymbol{\mu} \otimes \boldsymbol{I}_m) \qquad (10)$$

Clearly, Σ_i' is positive semi-definite and depends only on the ith element of data, its covariance $\Lambda_{\boldsymbol{x}_i}$, and the parameter vector $\boldsymbol{\mu}$. Assume henceforth that each Σ_i' is positive definite and hence invertible. The inverses $\Sigma_i'^{-1}$ can now be used as matricial weights to define a 'centroid' of the $\boldsymbol{Z}_i$ as follows:

$$\tilde{\boldsymbol{Z}} = \sum_{i=1}^{n} \boldsymbol{Z}_i \Sigma_i'^{-1} \left(\sum_{i=1}^{n} \Sigma_i'^{-1} \right)^{-1} \qquad (11)$$

Here, $\sum_{i=1}^{n} \Sigma_i'^{-1}$ is invertible because a sum of positive definite matrices is also positive definite. For each $i = 1, \ldots, n$, let

$$\boldsymbol{Z}_i' = \boldsymbol{Z}_i - \tilde{\boldsymbol{Z}} \qquad (12)$$

be the ith pure measurement vector relative to $\tilde{\boldsymbol{Z}}$. Letting

$$\boldsymbol{\eta}_i' = \Sigma_i'^{-1}\boldsymbol{Z}_i'^{\mathrm{T}}\boldsymbol{\mu} \qquad (13)$$

define the following $(l-m) \times (l-m)$ matrices

$$\boldsymbol{M}_{\boldsymbol{\mu}}' = \sum_{i=1}^{n} \boldsymbol{Z}_i' \Sigma_i'^{-1} \boldsymbol{Z}_i'^{\mathrm{T}} \qquad (14a)$$

$$\boldsymbol{N}_{\boldsymbol{\mu}}' = \sum_{i=1}^{n} (\boldsymbol{I}_{l-m} \otimes \boldsymbol{\eta}_i'^{\mathrm{T}})\boldsymbol{B}_i^0(\boldsymbol{I}_{l-m} \otimes \boldsymbol{\eta}_i') \qquad (14b)$$

$$\boldsymbol{X}_{\boldsymbol{\mu}}' = \boldsymbol{M}_{\boldsymbol{\mu}}' - \boldsymbol{N}_{\boldsymbol{\mu}}'$$

A fundamental result that can now be established is that $\boldsymbol{\theta} = [\boldsymbol{\mu}^{\mathrm{T}}, \boldsymbol{\alpha}^{\mathrm{T}}]^{\mathrm{T}}$ satisfies the variational equation (6) if and only if the following system of equations holds:

$$\boldsymbol{X}_{\boldsymbol{\mu}}'\boldsymbol{\mu} = \boldsymbol{0} \qquad (15a)$$

$$\boldsymbol{\alpha} = -(\tilde{\boldsymbol{Z}}\boldsymbol{W}^{-1})^{\mathrm{T}}\boldsymbol{\mu} \qquad (15b)$$

A proof can be found in Appendix 1. The first equation constrains solely $\boldsymbol{\mu}$ and therefore can be solved separately. Once $\boldsymbol{\mu}$ is determined, $\boldsymbol{\alpha}$ is readily prescribed by the second equation. Of the two constraints, the first plays a leading role and will be termed the reduced variational equation.

With the reduced AML cost function defined by

$$J_{\mathrm{AML}}'(\boldsymbol{\mu}; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{Z}_i'\Sigma_i'^{-1}\boldsymbol{Z}_i'^{\mathrm{T}}\boldsymbol{\mu} \qquad (16)$$

(15a) can be viewed as the variational equation for an optimiser of J_{AML}'. Interestingly, the $\boldsymbol{\mu}$-part of $\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^u$, which satisfies (15a) as $\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^u$ satisfies (6), turns out to be the minimiser of J_{AML}', denoted $\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^u$, not just a critical point of J_{AML}'. Moreover, both J_{AML} and J_{AML}' attain a common minimum value at $\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^u$ and $\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^u$, respectively (see Appendix 2). One noteworthy consequence of this link is that the reduced AML cost function can be minimised by any algorithm and the result (a $\boldsymbol{\mu}$-vector) can first be fed into (15b) to produce a partial estimate (an $\boldsymbol{\alpha}$-vector) and further combined with this partial estimate [as per (8)] to produce the minimiser of the full AML cost function.

4 FNS: full and reduced forms

A vector $\boldsymbol{\theta}$ satisfies (6) if and only if it is a solution of the ordinary eigenvalue problem $\boldsymbol{X}_{\boldsymbol{\theta}}\boldsymbol{\xi} = \lambda\boldsymbol{\xi}$ corresponding to the eigenvalue $\lambda = 0$. This suggests an iterative method for solving (6) whereby if $\boldsymbol{\theta}_{k-1}$ is a current approximate solution, then an updated solution $\boldsymbol{\theta}_k$ is a vector chosen from that eigenspace of $\boldsymbol{X}_{\boldsymbol{\theta}_{k-1}}$, which most closely approximates the null space of $\boldsymbol{X}_{\boldsymbol{\theta}}$; this eigenspace is, of course, the one corresponding to the eigenvalue closest to zero in absolute value. The process can be seeded with the normalised algebraic least squares (NALS) estimate, $\hat{\boldsymbol{\theta}}_{\mathrm{NALS}}$. This estimate results from applying the algebraic least-squares (ALS) method to Hartley-normalised data [15]. ALS is a simple technique that computes the unconstrained minimiser of the cost function $J_{\mathrm{ALS}}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|^{-2}\sum_{i=1}^{n}\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{U}_i\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{\theta}$, with $\|\boldsymbol{\theta}\| = (\Sigma_{j=1}^{l}\theta_j^2)^{1/2}$, by performing singular-value decomposition on $[\boldsymbol{U}_1, \ldots, \boldsymbol{U}_n]^{\mathrm{T}}$. The overall procedure constitutes the FNS [10, 11] and is summarised in Algorithm 1.

Algorithm 1: Steps of the FNS

1. Set $\boldsymbol{\theta}_0 = \hat{\boldsymbol{\theta}}_{\mathrm{NALS}}$.

2. Assuming $\boldsymbol{\theta}_{k-1}$ is known, compute the matrix $\boldsymbol{X}_{\boldsymbol{\theta}_{k-1}}$.

3. Compute a normalised eigenvector of $\boldsymbol{X}_{\boldsymbol{\theta}_{k-1}}$ corresponding to the eigenvalue closest to zero (in absolute value) and take this eigenvector for $\boldsymbol{\theta}_k$.

4. If $\boldsymbol{\theta}_k$ is sufficiently close to $\boldsymbol{\theta}_{k-1}$, then terminate the procedure; otherwise increment k and return to Step 2.

A modification of this technique based on the reduced variational system (15a) and (15b) is the reduced FNS (RFNS). Its steps are given in Algorithm 2.

Algorithm 2: Steps of the RFNS

1. Set $\boldsymbol{\mu}_0 = \hat{\boldsymbol{\mu}}_{\mathrm{NALS}}$.

2. Assuming $\boldsymbol{\mu}_{k-1}$ is known, compute the matrix $X'_{\boldsymbol{\mu}_{k-1}}$.

3. Compute a normalised eigenvector of $X'_{\boldsymbol{\mu}_{k-1}}$ corresponding to the eigenvalue closest to zero (in absolute value) and take this eigenvector for $\boldsymbol{\mu}_k$.

4. If $\boldsymbol{\mu}_k$ is sufficiently close to $\boldsymbol{\mu}_{k-1}$, then terminate the procedure; otherwise increment k and return to Step 2.

5. Compute $\boldsymbol{\alpha}$ as per (15b) using the limiting value $\boldsymbol{\mu}_k$ and the corresponding value $\tilde{Z}(\boldsymbol{\mu}_k)$ from the previous step.

In the case that the matrices Σ_i^{-1} are replaced by the matrices $(\Sigma_i)_r^+$ in the expression for J_{AML}, a similar change also affects the matrices $X_{\boldsymbol{\theta}_k}$ of FNS. Moreover, as $\Sigma_i = \Sigma_i'$ for $i = 1, \ldots, n$ (see Appendix 1), the $(\Sigma_i')_r^+$ supersede the $\Sigma_i'^{-1}$ in the expression for J'_{AML} and in the $X'_{\boldsymbol{\mu}_k}$ of RFNS. The limiting estimates produced by the modified FNS and the modified iterative part of RFNS are approximate solutions of the variational and reduced variational equations for the modified J_{AML} and J'_{AML}, respectively.

Finally, we remark that a vector $\boldsymbol{\theta}$ satisfying (6) can alternatively be viewed as a solution of the generalised eigenvalue problem $M_{\boldsymbol{\theta}}\boldsymbol{\xi} = \lambda N_{\boldsymbol{\theta}}\boldsymbol{\xi}$ corresponding to the eigenvalue $\lambda = 1$. This observation provides a starting point for the development of the HEIV scheme in both full and reduced versions [13]. Each version solves successively generalised eigenvalue problems analogous to the ordinary eigenvalue problems solved by a corresponding version of FNS.

5 Experimental evaluation

We now present the results of comparative tests carried out to evaluate the performance of both the RFNS and reduced RHEIV. The application considered is trifocal tensor estimation from point correspondences. The following six algorithms were used to compute tensors from both synthetic and real image data

- NALS = normalised algebraic least squares

- HEIV = heteroscedastic errors-in-variables scheme

- RHEIV = reduced HEIV scheme

- FNS = fundamental numerical scheme

- RFNS = reduced FNS

- GS = gold standard

GS is an advanced method [3] for minimising the maximum likelihood cost function J_{ML}, the expression for which is given in (17) below. For fair comparison, FNS, RFNS, RHEIV and GS are seeded with the generalised total least-squares tensor estimate [12, 16] in order to match HEIV given in [16]. A post-correction was applied to the final tensors obtained by FNS, RFNS and RHEIV to enforce internal constraints [16]. In the following, the symbol '+' is appended to the acronym of a given method to indicate that a correction process was operated upon the output of the method. HEIV estimates were obtained by using a binary application [17] supplied by Matei and Meer [5]. These estimates satisfy the internal constraints, as the application generating it uses a variant of the HEIV algorithm that dynamically enforces the constraints throughout the iterative process.

5.1 Trifocal tensor from point matches

A trifocal tensor of three views is an analogue of a fundamental matrix of two views. It encapsulates all the geometric relations between three views that are independent of scene structure, but is more useful than the fundamental matrix as it captures constraints not only on point correspondences but also on line and combined point-and-line correspondences across the images.

Suppose that three camera matrices are chosen as $\boldsymbol{P} = [\boldsymbol{I}_3, \boldsymbol{0}]$, $\boldsymbol{P}' = [\boldsymbol{A}, \boldsymbol{e}'] = [a_i^j]$, and $\boldsymbol{P}'' = [\boldsymbol{B}, \boldsymbol{e}''] = [b_i^k]$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are 3×3 matrices describing the infinite homographies from the first to the second and third images, respectively, and $\boldsymbol{e}'$ and $\boldsymbol{e}''$ are the epipoles in the latter two views. The trifocal tensor is the valence-3 tensor given by

$$\mathcal{J}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k, \quad i, j, k = 1, 2, 3$$

Let $\boldsymbol{m} = [m^1, m^2, m^3]^{\mathrm{T}}$, $\boldsymbol{m}' = [m'^1, m'^2, 1]^{\mathrm{T}}$ and $\boldsymbol{m}'' = [m''^1, m''^2, 1]^{\mathrm{T}}$ be the images of a point $\boldsymbol{M}$ in 3D space, taken from the cameras with corresponding superscripts. The points are related through the trifocal tensor by the four trilinear constraints [4]

$$\sum_{i=1}^3 (m^i \mathcal{J}_i^{11} - m^i m'^1 \mathcal{J}_i^{31} + m^i m'^1 m''^1 \mathcal{J}_i^{33} - m^i m''^1 \mathcal{J}_i^{13}) = 0$$

$$\sum_{i=1}^3 (m^i \mathcal{J}_i^{12} - m^i m'^1 \mathcal{J}_i^{32} + m^i m'^1 m''^2 \mathcal{J}_i^{33} - m^i m''^2 \mathcal{J}_i^{13}) = 0$$

$$\sum_{i=1}^3 (m^i \mathcal{J}_i^{21} - m^i m'^2 \mathcal{J}_i^{31} + m^i m'^2 m''^1 \mathcal{J}_i^{33} - m^i m''^1 \mathcal{J}_i^{23}) = 0$$

$$\sum_{i=1}^3 (m^i \mathcal{J}_i^{22} - m^i m'^2 \mathcal{J}_i^{32} + m^i m'^2 m''^2 \mathcal{J}_i^{33} - m^i m''^2 \mathcal{J}_i^{23}) = 0$$

Letting $m^3 = 1$, this system can be brought into the form given in (2) by first concatenating the inhomogeneous coordinates of $\boldsymbol{m}$, $\boldsymbol{m}'$, and $\boldsymbol{m}''$ to obtain a single item of data $\boldsymbol{x} = [m^1, m^2, m'^1, m'^2, m''^1, m''^2]$, next rearranging the tensor entries into a length-27 vector $\boldsymbol{\theta}$, and then setting $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) = [f_1(\boldsymbol{x}, \boldsymbol{\theta}), \dots, f_4(\boldsymbol{x}, \boldsymbol{\theta})]^T$, where $f_1, \dots, f_4$ are the corresponding expressions on the left-hand side of the above system. Furthermore, $\boldsymbol{U}(\boldsymbol{x})$ and $\boldsymbol{\theta}$ can be partitioned as in (7) and (8) with $\boldsymbol{W} = \boldsymbol{I}_4$ and $\boldsymbol{\alpha} = [\mathcal{J}_3^{11}, \mathcal{J}_3^{12}, \mathcal{J}_3^{21}, \mathcal{J}_3^{22}]^T$, respectively. Given a data set $\{\boldsymbol{x}_i\}_{i=1}^n$, the covariance of each datum $\boldsymbol{x}_i$ is assumed to be the default 6×6 identity matrix corresponding to isotropic homogeneous noise in image point measurement. Lastly, since the submanifolds of the form $\{\boldsymbol{x} \in \mathbb{R}^6 | \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{0}\}$ with $\boldsymbol{\theta}$ representing a genuine trifocal tensor, have codimension 3 [18], the matrices $(\Sigma_i)_3^+$ and $(\Sigma'_i)_3^+$ are used instead of the matrices Σ_i^{-1} and $\Sigma_i'^{-1}$ in the expressions for J_{AML} and J'_{AML}, respectively, and in related corresponding entities.

5.2 Synthetic image tests

First, the performance of the estimators was evaluated in a series of 200 tests on synthetic input. A set of 3D points was created in a cuboid of dimensions 3 m × 1.5 m × 3 m with five points equally spaced along each direction. The images were 3000 × 2000 pixels in size, with square pixels of side 9 μm. The 125 world points were captured by three perspective cameras with focal length of 3600 pixels, placed at $\tilde{\boldsymbol{C}}_1 = [-5, 3, 1.5]^T$, $\tilde{\boldsymbol{C}}_2 = [0, 0, 0]^T$, and $\tilde{\boldsymbol{C}}_3 = [3, 3, 1.5]^T$ to provide 'true' matches. The centre of the cuboid was located 5 m away from the world origin at $\tilde{\boldsymbol{C}}_2$. Fig. 1 depicts the 3D scene with the camera positions and orientations. Each true image point was then perturbed by homogeneous Gaussian noise with zero mean and standard deviation of $\sigma = 2$ pixels. The resulting noise-contaminated triples of matching points were used as input to the six algorithms.
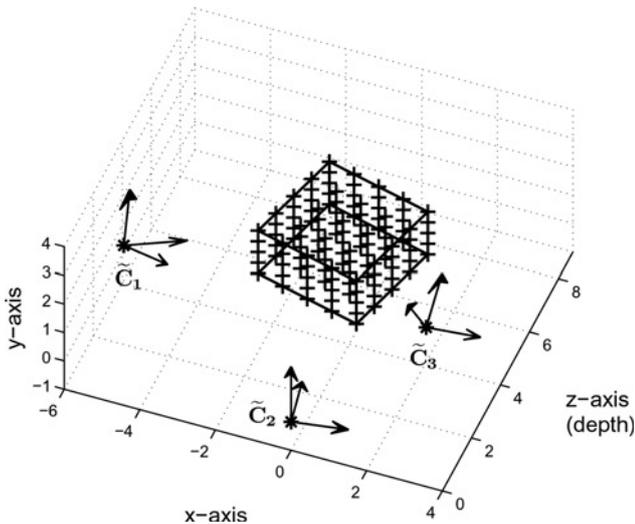

Figure 1 *A synthetic 3D scene made of equally spaced points inside a cuboid and three cameras viewing the scene*

Table 1 shows the averages over the total number of trials for the unconstrained schemes. To calculate J'_{AML} residuals for the non-reduced methods, the $\boldsymbol{\mu}$-component of each final $\boldsymbol{\theta}$-vector was retrieved and plugged into the J'_{AML} expression. It is clearly seen that the estimates produced by the iterative schemes give all similar values of J_{AML} and J'_{AML} residuals. This, in particular, provides an empirical confirmation of the identity $J_{AML}(\boldsymbol{\theta}) = J'_{AML}(\boldsymbol{\mu})$ (see Appendix 2). RFNS and RHEIV achieved a better convergence rate over FNS − the last scheme was almost three times slower and required about three times as many iterations compared with the two other schemes. It can also be seen that the iterative methods are tangibly more accurate than the basic non-iterative NALS procedure. This suggests potential utility of the iterative methods, as only accurate unconstrained estimates can be upgraded to accurate constrained estimates suitable for practical applications.

Table 2 shows the averages for the constrained schemes with the addition of two other tests. It is immediately apparent that the J_{AML} and J'_{AML} values for the HEIV, RHEIV+, FNS+, and RFNS+ estimates are all very similar, if slightly higher than those for the estimates generated by the respective unconstrained schemes. Moreover, in terms of J_{AML} and J'_{AML} values, the estimates produced by all four constrained methods are of comparable accuracy to the GS estimates.

The third and fourth columns give the results of perhaps more critical tests coming from using the maximum likelihood function, J_{ML}. For a trifocal tensor estimate $\hat{\boldsymbol{\theta}}$ obtained by a method other than GS, $J_{ML}(\hat{\boldsymbol{\theta}})$ was calculated by minimising the reprojection error

$$\sum_{i=1}^n \left(d(\boldsymbol{m}_i, \hat{\boldsymbol{m}}_i)^2 + d(\boldsymbol{m}'_i, \hat{\boldsymbol{m}}'_i)^2 + d(\boldsymbol{m}''_i, \hat{\boldsymbol{m}}''_i)^2 \right) \quad (17)$$

over all points $\hat{\boldsymbol{m}}_i = \mathcal{N}(\boldsymbol{P}\boldsymbol{M}_i)$, $\hat{\boldsymbol{m}}'_i = \mathcal{N}(\hat{\boldsymbol{P}}'\boldsymbol{M}_i)$ and $\hat{\boldsymbol{m}}''_i = \mathcal{N}(\hat{\boldsymbol{P}}''\boldsymbol{M}_i)$, where $\hat{\boldsymbol{P}}'$ and $\hat{\boldsymbol{P}}''$ are retrieved from $\hat{\boldsymbol{\theta}}$ [3] and kept fixed. Note the difference with the GS algorithm, in which − for finding $\hat{\boldsymbol{\theta}}$ that minimises the reprojection error − the $\hat{\boldsymbol{m}}_i$, $\hat{\boldsymbol{m}}'_i$, $\hat{\boldsymbol{m}}''_i$, and $\hat{\boldsymbol{P}}'$ and $\hat{\boldsymbol{P}}''$ are allowed to vary simultaneously. Here, $\mathcal{N}(\boldsymbol{m}) = \boldsymbol{m}/m^3$ is a normalisation procedure whose application ensures that the third coordinate of a given planar point is unity, and $d(\boldsymbol{m}, \boldsymbol{n})$ denotes the

Table 1 Average J_{AML} and J'_{AML} residuals, number of iterations, and timing for the unconstrained schemes

Methods	J_{AML}	J'_{AML}	Iter.	Time, s
NALS	1450.2	1450.1	1	0.05
FNS	1432.1	1432.1	9.4	2.50
RFNS	1432.1	1432.1	2.6	0.90
RHEIV	1432.1	1432.1	2.6	0.90

Table 2 Average J_{AML}, J'_{AML}, J_{ML} residuals, RMS error, number of iterations and timing for the constrained schemes

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time, s
HEIV	1462.6	1462.6	1463.0	1.40	–	0.01
RHEIV+	1462.6	1462.6	1462.6	1.40	2.6	1.49
FNS+	1462.6	1462.6	1462.6	1.40	9.4	2.69
RFNS+	1462.6	1462.6	1462.6	1.40	2.6	1.49
GS	1462.9	1462.8	1468.6	1.40	13.2	20.72

Euclidean distance between the planar points m and n that have been normalised in the above sense. The M_i are initially obtained by triangulating from the m_i, m'_i and m''_i and are then recomputed in each optimisation step of an iterative scheme (typically, and in our case, the Levenberg–Marquardt algorithm) that minimises the reprojection error. The root-mean-square (RMS) reprojection error is taken to be $(J_{ML}(\hat{\boldsymbol{\theta}})/(6n))^{1/2}$, with 6 representing the number of
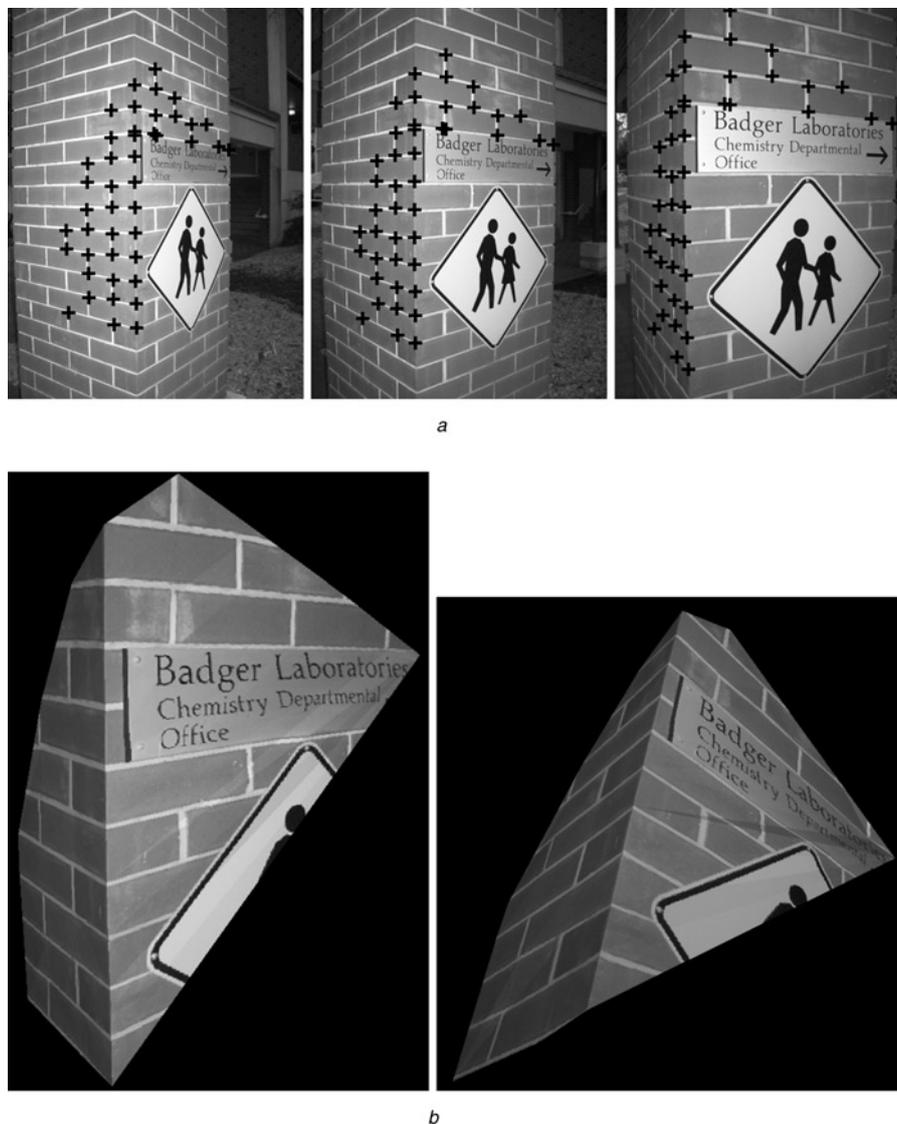

Figure 2 *Three images and a 3D model built from them by using a trifocal tensor estimate*

a Original: Each image is 600×800 pixels in size with 44 putative points identified
b Reconstruction : Two views of the 3D model obtained from RFNS+ trifocal tensor estimate

elementary degrees of freedom expressible in units of length: three images × two image dimensions. Upon inspection, it is found that the estimates from the post-corrected schemes produce very competitive J_{ML} and RMS error values in comparison to GS estimates, and that the post-corrected schemes achieve the same accuracy as HEIV, which matches expectations.

The last two columns of Table 2 provide insight as to the computing efficiency of the schemes. Here, the iteration number field for HEIV is left empty because the binary application used to generate HEIV estimates does not allow for determining the number of iterations involved. It is clear that FNS+, RFNS+, and RHEIV+ operate very quickly compared with GS. Note that the search space for GS has dimension $125 \times 3 + 27 = 402$, whereas the search space for FNS+ and that for RFNS+ and RHEIV+ have dimensions 27 and 23, respectively.

Finally, it should be pointed out that in another series of 200 tests, FNS failed to converge 27 times, whereas RFNS succeeded every time. This indicates that the eigenvalue problem solved by RFNS is better conditioned than the one solved by FNS.

5.3 Real image test

Three images were acquired by a hand-held camera and registered using a trifocal tensor computed with RFNS+ to build a 3D model (Fig. 2). Owing to a small baseline distance between any two camera positions, the trifocal plane here is not firmly defined and has potential to trigger numerical instabilities. To ensure convergence, a modification of FNS in Step 3 became necessary. With $\boldsymbol{v}_{i,k}$ the normalised eigenvector corresponding to the ith smallest eigenvalue of $\boldsymbol{X}_{\boldsymbol{\theta}_{k-1}}$, the update $\boldsymbol{\theta}_k$ was defined as the result of normalising $\sum_{i=1}^{3} (\boldsymbol{\theta}_{k-1}^{T} \boldsymbol{v}_{i,k}) \boldsymbol{v}_{i,k}$. A similar adjustment was made for RFNS.

Tables 3 and 4 show the results of applying the previous six algorithms to the image data points. As before, it can be seen that the unconstrained algorithms produce estimates with very similar J_{AML} and J'_{AML} residuals. FNS and RFNS converged in the same number of iterations and executed in about the same time. RHEIV lagged fractionally behind.

Tables 4 gives the results for the constrained schemes. All methods performed well and were essentially inseparable

Table 3 J_{AML} and J'_{AML} residuals, number of iterations and timing for the unconstrained schemes

Methods	J_{AML}	J'_{AML}	Iter.	Time, s
NALS	33.2	33.2	1	0.06
FNS	28.5	28.3	6	0.62
RFNS	28.6	28.6	6	0.64
RHEIV	28.7	28.6	8	0.72

Table 4 J_{AML}, J'_{AML}, J_{ML} residuals, RMS error, number of iterations and timing for the constrained schemes

Methods	J_{AML}	J'_{AML}	J_{ML}	RMS	Iter.	Time, s
HEIV	37.3	37.2	37.2	0.38	–	0.06
RHEIV+	37.3	37.2	37.3	0.38	8	0.83
FNS+	37.2	37.2	37.2	0.37	6	0.80
RFNS+	37.2	37.2	37.2	0.37	6	0.72
GS	37.2	37.2	37.2	0.37	233	37.89

from GS. Again, because of a much larger search space, GS ran markedly slower than the other methods, with RFNS+ being the fastest.

6 Conclusion

A novel parameter estimation method, RFNS, was proposed for a class of problems in which the relationship between parameters and image data is expressed as a system of equations. The original FNS method operates over the entire parameter space, whereas the newly proposed method operates on a subspace of lower dimension and recuperates the missing degrees of freedom in a single final step. The performance of RFNS was demonstrated on the problem of trifocal tensor estimation. It was shown that RFNS is more stable and faster to converge than FNS, the former scheme being the fastest of all the methods tested. It was also shown that when compared with GS, RFNS gives almost identical results in terms of the J_{AML} and J'_{AML} residuals, GS's reprojection error, and RMS error, indicating that RFNS produces a solution of high accuracy. Finally, it was noted that a companion scheme, RHEIV, can be evolved in a similar fashion to that of RFNS and it was found that the performance of RHEIV matches that of RFNS.

7 Acknowledgment

8 References

[1] ZOGHLAMI I., FAUGERAS O., DERICHE R.: 'Using geometric corners to build a 2D mosaic from a set of images'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1997, pp. 420–425

[2] FAUGERAS O., LUONG Q.-T., PAPADOPOULO T.: 'The Geometry of Multiple Images' (MIT Press, Cambridge, MA, 2001)

[3] HARTLEY R., ZISSERMAN A.: 'Multiple view geometry in computer vision' (Cambridge University Press, 2000)

[4] SHASHUA A.: 'Trilinear tensor: The fundamental construct of multiple-view geometry and its applications'. Proc. Int. Workshop Algebraic Frames for The Perception-Action Cycle, (*LNCS*, **1315**), Springer, 1997, pp. 190−206

[5] MATEI B., MEER P.: 'Estimation of nonlinear errors-in-variables models for computer vision applications', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (10), pp. 1537−1552

[6] CHOJNACKI W., VAN DEN HENGEL A., BROOKS M.J.: 'Generalised principal component analysis: Exploiting inherent parameter constraints, in BRAZ J., RANCHORDAS A., ARAÚJO H., JORGE J. (EDS.): 'Advances in Computer Graphics and Computer Vision' (*Communications in Computer and Information Science*, **4**), Springer, 2008, pp. 217−228

[7] VIDAL R., MA Y., SASTRY S.: 'Generalized principal component analysis (GPCA)', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (12), pp. 1945−1959

[8] ENGL H.W., HANKE M., NEUBAUER A.: 'Regularization of Inverse Problems' (Kluwer, Dordrecht, 1996)

[9] KANATANI K.: 'Statistical optimization for geometric computation: theory and practice' (Elsevier, Amsterdam, 1996)

[10] SCOLERI T., CHOJNACKI W., BROOKS M.J.: 'A multi-objective parameter estimator for image mosaicing'. Proc. IEEE Int. Symposium Signal Processing and its Applications, 2005, pp. 551−554

[11] CHOJNACKI W., BROOKS M.J., VAN DEN HENGEL A., GAWLEY D.: 'On the fitting of surfaces to data with covariances', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, **22**, (11), pp. 1294−1303

[12] LEEDAN Y., MEER P.: 'Heteroscedastic regression in computer vision: problems with bilinear constraint', *Int. J. Comput. Vision*, 2000, **37**, (2), pp. 127−150

[13] CHOJNACKI W., BROOKS M.J., VAN DEN HENGEL A., GAWLEY D.: 'From FNS to HEIV: A link between two vision parameter estimation methods', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004, **26**, (2), pp. 264−268

[14] LÜTKEPOL H.: 'Handbook of matrices' (John Wiley & Sons, Chichester, 1996)

[15] HARTLEY R.: 'Lines and points in three views and the trifocal tensor', *Int. J. Comput. Vision*, 1997, **22**, (2), pp. 125−140

[16] MATEI B., MEER P.: 'A versatile method for trifocal tensor estimation'. Proc. Eighth Int. Conf. Computer Vision, 2001, vol. 2, pp. 578−585

[17] http://www.caip.rutgers.edu/riul/research/code.html

[18] TORR P.H.S., ZISSERMAN A.: 'Robust parametrization and computation of the trifocal tensor', *Image and Vision Comput.*, 1997, **15**, pp. 591−605

9 Appendixes

9.1 Appendix 1

Here, we show the equivalence of (6) and the system comprising (15a) and (15b). Recalling definitions (5d) and (10), first note that, by (8) and (9)

$$\Sigma_i = \Sigma_i' \tag{18}$$

for each $i = 1, \ldots, n$. Consequently, definition (5a) can be rephrased as

$$M_{\boldsymbol{\theta}} = \sum_{i=1}^{n} U_i \Sigma_i'^{-1} U_i^{\mathrm{T}} \tag{19}$$

Again by (9), for each $i = 1, \ldots, n$

$$
(I_l \otimes \boldsymbol{\eta}_i^{\mathrm{T}}) B_i (I_l \otimes \boldsymbol{\eta}_i) = \begin{bmatrix} I_{l-m} \otimes \boldsymbol{\eta}_i^{\mathrm{T}} & \mathbf{0}_{(l-m) \times m^2} \\ \mathbf{0}_{m \times m(l-m)} & I_m \otimes \boldsymbol{\eta}_i^{\mathrm{T}} \end{bmatrix}
$$
$$
\times \begin{bmatrix} B_i^0 & \mathbf{0}_{m(l-m) \times m^2} \\ \mathbf{0}_{m^2 \times m(l-m)} & \mathbf{0}_{m^2 \times m^2} \end{bmatrix}
$$
$$
\times \begin{bmatrix} I_{l-m} \otimes \boldsymbol{\eta}_i & \mathbf{0}_{m(l-m) \times m} \\ \mathbf{0}_{m^2 \times (l-m)} & I_m \otimes \boldsymbol{\eta}_i \end{bmatrix}
$$
$$
= \begin{bmatrix} (I_{l-m} \otimes \boldsymbol{\eta}_i^{\mathrm{T}}) B_i^0 (I_{l-m} \otimes \boldsymbol{\eta}_i) & \mathbf{0}_{(l-m) \times m} \\ \mathbf{0}_{m \times (l-m)} & \mathbf{0}_{m \times m} \end{bmatrix}
$$

It follows that $N_{\boldsymbol{\theta}}$ given in (5b) takes the form

$$N_{\boldsymbol{\theta}} = \begin{bmatrix} N_{\boldsymbol{\theta}}^0 & \mathbf{0}_{(l-m) \times m} \\ \mathbf{0}_{m \times (l-m)} & \mathbf{0}_{m \times m} \end{bmatrix} \tag{20}$$

where

$$N_{\boldsymbol{\theta}}^0 = \sum_{i=1}^{n} (I_{l-m} \otimes \boldsymbol{\eta}_i^{T}) B_i^0 (I_{l-m} \otimes \boldsymbol{\eta}_i) \tag{21}$$

Now, if $\boldsymbol{\theta}$ satisfies (6), then, in view of (7), (19) and (20), the equivalent condition $M_{\boldsymbol{\theta}} \boldsymbol{\theta} = N_{\boldsymbol{\theta}} \boldsymbol{\theta}$ on $\boldsymbol{\theta}$ can be written as

$$\sum_{i=1}^{n} \begin{bmatrix} Z_i \\ W \end{bmatrix} \Sigma_i'^{-1} [Z_i^{\mathrm{T}}, W^{\mathrm{T}}] \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} N_{\boldsymbol{\theta}}^0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\alpha} \end{bmatrix}$$

which in turn expands into the system

$$\sum_{i=1}^{n} Z_i \Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = N_\theta^0 \mu \qquad (22a)$$

$$\sum_{i=1}^{n} W \Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = 0 \qquad (22b)$$

By our standing assumption that W is invertible, the second of the above equations reduces to

$$\sum_{i=1}^{n} \Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = 0 \qquad (23)$$

Now, since the Σ_i' and hence the $\Sigma_i'^{-1}$ are symmetric, it immediately follows from (11) that $\tilde{Z}^{\mathrm{T}} = (\sum_{i=1}^{n} \Sigma_i'^{-1})^{-1} \sum_{i=1}^{n} \Sigma_i'^{-1} Z_i^{\mathrm{T}}$. Hence, (23) can be rewritten as

$$\left(\sum_{i=1}^{n} \Sigma_i'^{-1}\right)(W^{\mathrm{T}}\alpha + \tilde{Z}^{\mathrm{T}}\mu) = 0$$

and further as

$$W^{\mathrm{T}}\alpha + \tilde{Z}^{\mathrm{T}}\mu = 0 \qquad (24)$$

As W is invertible, this immediately implies (15b).

To show that (15a) also holds, note that, by (7) and (8), for each $i = 1, \ldots, n$

$$U_i^{\mathrm{T}}\theta = W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu$$

and by (12) and (24)

$$W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu = W^{\mathrm{T}}\alpha + (Z_i'^{\mathrm{T}} + \tilde{Z}^{\mathrm{T}})\mu = Z_i'^{\mathrm{T}}\mu$$

whence

$$U_i^{\mathrm{T}}\theta = Z_i'^{\mathrm{T}}\mu \qquad (25)$$

Recalling definitions (5e) and (13), we see that (25) combined with (18) implies that $\eta_i = \eta_i'$. Comparison of (14b) and (21) now yields $N_\theta^0 = N_\mu'$. Thus, in particular

$$N_\theta^0 \mu = N_\mu' \mu \qquad (26)$$

Furthermore, in view of (12)

$$\sum_{i=1}^{n} Z_i \Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = \sum_{i=1}^{n}(Z_i' + \tilde{Z})\Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu)$$

By (23)

$$\sum_{i=1}^{n} \tilde{Z}\Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = \tilde{Z}\sum_{i=1}^{n}\Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = 0$$

and by (12), (14a) and (24)

$$\sum_{i=1}^{n} Z_i' \Sigma_i'^{-1}(W^{\mathrm{T}}\alpha + Z_i^{\mathrm{T}}\mu) = \sum_{i=1}^{n} Z_i' \Sigma_i'^{-1}$$

$$\times (W^{\mathrm{T}}\alpha + \tilde{Z}^{\mathrm{T}}\mu + Z_i'^{\mathrm{T}}\mu) = \sum_{i=1}^{n} Z_i' \Sigma_i'^{-1} Z_i'^{\mathrm{T}}\mu = M_\mu'\mu$$

Putting the last three expressions together, we see that the left-hand side of (22a) is equal to $M_\mu'\mu$. This jointly with (26) yields (15a), as required.

Working backwards, one can easily infer that if μ and α satisfy (15a) and (15b) respectively, then $\theta = [\mu^{\mathrm{T}}, \alpha^{\mathrm{T}}]^{\mathrm{T}}$ satisfies the original expression (6).

9.2 Appendix 2

Let $\mu_{\hat{\theta}_{\mathrm{AML}}^u}$ and $\alpha_{\hat{\theta}_{\mathrm{AML}}^u}$ be the parts of $\hat{\theta}_{\mathrm{AML}}^u$ as per (8). Here, we show that $\mu_{\hat{\theta}_{\mathrm{AML}}^u}$ can be identified with $\hat{\mu}_{\mathrm{AML}}^u$, and, moreover, that both J_{AML} and J_{AML}' attain a common minimum value at $\hat{\theta}_{\mathrm{AML}}^u$ and $\hat{\mu}_{\mathrm{AML}}^u$, respectively.

First note that, in view of (18), the expression for J_{AML} given by

$$J_{\mathrm{AML}}(\theta) = \sum_{i=1}^{n} \theta^{\mathrm{T}} U_i \Sigma_i^{-1} U_i^{\mathrm{T}} \theta$$

can be restated as

$$J_{\mathrm{AML}}(\theta) = \sum_{i=1}^{n} \theta^{\mathrm{T}} U_i \Sigma_i'^{-1} U_i^{\mathrm{T}} \theta \qquad (27)$$

Next, given an arbitrary μ, let α be such that (15b) holds, and let $\theta = [\mu^{\mathrm{T}}, \alpha^{\mathrm{T}}]^{\mathrm{T}}$. Then, as the calculation in Appendix 1 immediately preceding (25) reveals, (25) holds, and this equality combined with (16) and (27) yields

$$J_{\mathrm{AML}}(\theta) = J_{\mathrm{AML}}'(\mu) \qquad (28)$$

Since $J_{\mathrm{AML}}(\hat{\theta}_{\mathrm{AML}}^u) \leq J_{\mathrm{AML}}(\theta)$, we see that $J_{\mathrm{AML}}(\hat{\theta}_{\mathrm{AML}}^u) \leq J_{\mathrm{AML}}'(\mu)$, and since μ can, in particular, be taken to be $\hat{\mu}_{\mathrm{AML}}^u$, we have

$$J_{\mathrm{AML}}(\hat{\theta}_{\mathrm{AML}}^u) \leq J_{\mathrm{AML}}'(\hat{\mu}_{\mathrm{AML}}^u) \qquad (29)$$

On the other hand, as (15b) holds for $\mu_{\hat{\theta}_{\mathrm{AML}}^u}$ and $\alpha_{\hat{\theta}_{\mathrm{AML}}^u}$ [recall that $\hat{\theta}_{\mathrm{AML}}^u$ satisfies (6), which, as shown earlier in Appendix 1, implies (15b)], (28) can be explicitly written

in this case as

$$J_{\mathrm{AML}}(\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u}) = J'_{\mathrm{AML}}(\boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u}}) \qquad (30)$$

But $J'_{\mathrm{AML}}(\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^{u}) \leq J'_{\mathrm{AML}}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu}$, and so, in particular

$$J'_{\mathrm{AML}}(\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^{u}) \leq J'_{\mathrm{AML}}(\boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u}}) \qquad (31)$$

Putting (29), (30) and (31) together, we obtain

$$J_{\mathrm{AML}}(\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^{u}) = J'_{\mathrm{AML}}(\boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u}}) = J_{\mathrm{AML}}(\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u})$$

Hence, it first follows that $\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^{u}$ is equal to $\boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u}}$ (up to scale), as, generically, the minimiser of J'_{AML} is uniquely defined (up to scale). Furthermore, we see that J_{AML} and J'_{AML} attain a common minimum value at $\hat{\boldsymbol{\theta}}_{\mathrm{AML}}^{u}$ and $\hat{\boldsymbol{\mu}}_{\mathrm{AML}}^{u}$, respectively. Our claims have thus been established.