Photogrammetric Record, 17(9#): 000–000 (April / October 200#)

BUNDLE ADJUSTMENT WITH AND WITHOUT DAMPING

NICLAS BÖRLIN (niclas.borlin@cs.umu.se)
Umeå University, Sweden

PIERRE GRUSSENMEYER (pierre.grussenmeyer@insa-strasbourg.fr)
Institut National des Sciences Appliquées de Strasbourg, France

Abstract

The least squares adjustment (LSA) method is studied as an optimisation problem and shown to be equivalent to the undamped Gauss-Newton (GN) optimisation method. Three problem-independent damping modifications of the GN method are presented: the line-search method of Armijo (GNA); the Levenberg-Marquardt algorithm (LM); and Levenberg-Marquardt with Powell dogleg (LMP). Furthermore, an additional problem-specific "veto" damping technique, based on the chirality condition, is suggested.

In a perturbation study on a terrestrial bundle adjustment problem the GNA and LMP methods with veto damping can increase the size of the pull-in region compared to the undamped method; the LM method showed less improvement. The results suggest that damped methods can, in many cases, provide a solution where undamped methods fail and should be available in any LSA software package.

Matlab code for the algorithms discussed is available from the authors.

KEYWORDS: bundle, adjustment, least squares, convergence, initial values, terrestrial photogrammetry

INTRODUCTION

ONE IMPORTANT TASK in photogrammetry is to derive 3D information from 2D measurements. This is often performed using a bundle adjustment (BA) process. In a general sense, a BA problem may be considered to consist of the following nine elements:

(1) A *projection model* that describes the projection of a 3D object point (OP) into 2D image points (IP) in images taken by a camera. The internal geometry of the camera is described by its interior orientation (IO) parameters, such as the principal distance (camera constant), principal point and lens distortion parameters. The position and orientation of the camera in the object space is described by its exterior orientation (EO) parameters.

(2) A set of unknown parameters to be determined. This can be any combination of IO, EO and OP parameters.

(3) A number of observations, typically IP measurements, but may also consist of observations of camera positions, for instance.
(4) A set of known parameters, such as IO parameters or control point (CP) coordinates.
(5) Optionally a set of constraints between the parameters and/or the observations.
(6) A method to find initial values for the unknown parameters.
(7) An adjustment algorithm to find optimal parameter values based on the initial values.
(8) A method to automatically detect and exclude outliers in the observations.
(9) A method to automatically detect and exclude unstable parameters (those that can only be determined with a very low precision).

Aim of paper and related work

In this paper we focus strictly on step (7): adjustment algorithms to find optimal values based on given initial values of the unknown parameters. The aim of this paper is to compare the convergence properties of different least squares adjustment algorithms, especially with respect to their 'pull-in' radius applied to the BA problem. This has previously been studied within photogrammetry by, for example, Jianchar and Chern (2001) on the space resection problem and by Börlin et al. (2004) and Lourakis and Argyros (2005) on the BA problem.

LEAST SQUARES ADJUSTMENT

Introduction

Least squares adjustment (LSA) is a general method for estimating parameters from observations. LSA consists of a functional model and a stochastic model. The functional model describes the relationship between the parameters and observations in the absence of errors, such as the projection of a 3D object point (unknown parameter) onto 2D image pointrs (observations) through a (possibly unknown) camera perspective centre. The stochastic model describes the statistical properties of the observations, such as they are normally distributed with unknown mean and known covariance. The purpose of LSA is to find estimates of the unknown parameters based on the available observations and the functional and statistical models. The estimate should be unbiased and have minimum variance. If the functional model is non-linear and no direct solution exists, an iteration scheme is required in order to find the optimal solution. LSA is treated in numerous textbooks on photogrammetry and geodesy, such as Kraus (1993), Strang and Borre (1997) and McGlone et al. (2004). For this paper, Förstner and Wrobel (2004) is used as the main reference.

The Gauss-Markov non-linear functional model

The Gauss-Markov (GM) non-linear functional model is formulated as

$$\hat{\mathbf{l}} = \mathbf{l} + \hat{\mathbf{v}} = \mathbf{f}(\hat{\mathbf{p}}), \qquad (1)$$

where the vector **l** contains *m* observations, $\hat{\mathbf{v}}$ are the estimated residuals, and $\hat{\mathbf{p}}$ are the *n* estimated unknowns. The function **f** is the model function that describes the theoretical, potentially non-linear, relationship between the unknowns and the observations.

The linear substitute model

In general, it is not possible to use the non-linear model to directly estimate the unknowns and residuals. Instead the non-linear model is linearised to

$$\begin{gathered}\widehat{\Delta\mathbf{l}}_0 = \Delta\mathbf{l}_0 + \hat{\mathbf{v}}_0 = \mathbf{A}_0\widehat{\Delta\mathbf{p}}_0,\\ \text{where}\\ \Delta\mathbf{l}_0 = \mathbf{l} - \mathbf{f}(\hat{\mathbf{p}}_0),\\ \widehat{\Delta\mathbf{p}}_0 = \hat{\mathbf{p}} - \hat{\mathbf{p}}_0,\end{gathered} \tag{2}$$

where $\hat{\mathbf{p}}_0$ is the initial estimate of the unknowns and $\mathbf{A}_0$ is the design matrix – the Jacobian of the model function **f** with respect to the unknowns **p**. If **f** is non-linear, the design matrix $\mathbf{A}_0$ will be non-constant and must be recalculated at each iteration based on the current parameter estimates.

The stochastic model

The observations **l** are assumed to have a covariance matrix $\sigma_0^2\mathbf{C}_{\mathbf{ll}}$, where the covariance structure matrix $\mathbf{C}_{\mathbf{ll}}$ is known and has full rank. The variance of unit weight σ_0^2 is possibly unknown.

Estimation

Based on the stochastic model presented above, the optimal estimate $\hat{\mathbf{p}}$ of the unknowns is found by minimising the quadratic form

$$\Omega^2 = \hat{\mathbf{v}}^{\mathrm{T}}\mathbf{C}_{\mathbf{ll}}^{-1}\hat{\mathbf{v}} = \left(\hat{\mathbf{l}} - \mathbf{l}\right)^{\mathbf{T}}\mathbf{C}_{\mathbf{ll}}^{-1}\left(\hat{\mathbf{l}} - \mathbf{l}\right). \tag{3}$$

Substituting the linear model (2), we find the normal equations

$$\mathbf{A}_0^{\mathrm{T}}\mathbf{C}_{\mathbf{ll}}^{-1}\mathbf{A}_0\widehat{\Delta\mathbf{p}}_0 = \mathbf{A}_0^{\mathrm{T}}\mathbf{C}_{\mathbf{ll}}^{-1}\widehat{\Delta\mathbf{l}}_0. \tag{4}$$

The estimate of the covariance of the estimated parameters based on the linear model is

$$\hat{\mathbf{C}}_{\hat{\mathbf{p}}\hat{\mathbf{p}}} = \widehat{\sigma_0}^2\mathbf{C}_{\hat{\mathbf{p}}\hat{\mathbf{p}}}, \text{where} \tag{5}$$

$$\mathbf{C}_{\hat{\mathbf{p}}\hat{\mathbf{p}}} = (\mathbf{A}_0^{\mathrm{T}}\mathbf{C}_{\mathbf{ll}}^{-1}\mathbf{A}_0)^{-1}. \tag{6}$$

The variance factor $\widehat{\sigma_0}^2$ is estimated from the a posteriori residuals $\hat{\mathbf{v}}$ as

$$\widehat{\sigma_0}^2 = \frac{\hat{\mathbf{v}}^{\mathrm{T}}\mathbf{C}_{\mathbf{ll}}^{-1}\hat{\mathbf{v}}}{m-n}, \text{where} \tag{7}$$

$$\hat{\mathbf{v}} = \Delta\mathbf{l}_0 - \mathbf{A}_0\widehat{\Delta\mathbf{p}}_0. \tag{8}$$

Iteration scheme

ALGORITHM 1: *Gauss-Markov least squares adjustment*

```
1. Start with initial values $\hat{\mathbf{p}}_0$of the parameters and $k = 0$ ($k$ is the iteration number).
2. Repeat for $k = 1, 2, \ldots$ “until convergence” or $k > k_{\max}$.
   2.1. Calculate the observation update vector $\widehat{\Delta\mathbf{l}}_k = \mathbf{l} - \mathbf{f}(\hat{\mathbf{p}}_k)$ and design matrix
        $\mathbf{A}_k$ based on the current estimates $\hat{\mathbf{p}}_k$of the unknowns.
```

2.2. Solve the normal equations (4) for $\widehat{\Delta\mathbf{p}}_k$.

2.3. Update the estimated parameters

$$\widehat{\mathbf{p}}_{k+1} = \widehat{\mathbf{p}}_k + \widehat{\Delta\mathbf{p}}_k. \tag{9}$$

In principle, the estimated observations are also updated as

$$\hat{\mathbf{l}}_{k+1} = \mathbf{l} + \hat{\mathbf{v}}_k. \tag{10}$$

However, for the GM model we can instead use equation (1)

$$\hat{\mathbf{l}}_{k+1} = \mathbf{f}(\widehat{\mathbf{p}}_{k+1}). \tag{11}$$

The LSA algorithm for the GM model is summarised as Alg. 1.

NON-LINEAR OPTIMISATION

The focus of non-linear optimisation is to find the minimiser of a mathematical function – called the *objective function*. In this paper we will focus on non-linear least squares problems. For a more general treatment, consider for instance Nocedal and Wright (2006).

The weighted least squares problem and the Gauss-Newton method

A general weighted least squares optimisation problem may be written as

$$\min_{\mathbf{x}} F(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2}\mathbf{r}(\mathbf{x})^{\mathrm{T}}\mathbf{W}\mathbf{r}(\mathbf{x}), \tag{12}$$

where $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^m$is a vector-valued residual function, $\mathbf{W}$ is a positive semidefinite weight matrix, and m is the number of observations. With $\mathbf{W} = \mathbf{C}_{\mathbf{ll}}^{-1}$, we identify the objective function $F(\mathbf{x})$ with the quadratic form in equation (3).

Given an approximation $\mathbf{x}_k$ of the minimiser, consider the linear approximation of the residual function

$$\mathbf{r}(\mathbf{x}_k + \mathbf{s}) \approx \mathbf{r}_k + \mathbf{J}_k\mathbf{s}, \tag{13}$$

where $\mathbf{r}_k = \mathbf{r}(\mathbf{x}_k)$ and $\mathbf{J}_k = \mathbf{J}(\mathbf{x}_k)$ is the *Jacobian* of the residual function evaluated at the current approximation $\mathbf{x}_k$. If we formulate the minimisation problem as

$$\min_{\mathbf{s}_k} \frac{1}{2}\|\mathbf{r}_k + \mathbf{J}_k\mathbf{s}_k\|_{\mathbf{W}}^2 = \min_{\mathbf{s}_k} \frac{1}{2}(\mathbf{r}_k + \mathbf{J}_k\mathbf{s}_k)^{\mathrm{T}}\mathbf{W}(\mathbf{r}_k + \mathbf{J}_k\mathbf{s}_k), \tag{14}$$

we obtain the normal equations

$$\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{J}_k\mathbf{s}_k = \mathbf{J}_k^{\mathrm{T}}\mathbf{W}(-\mathbf{r}_k). \tag{15}$$

Substituting the Jacobian $\mathbf{J}$ with the design matrix $\mathbf{A}$ and the negative residual $-\mathbf{r}$ with the observation update $\widehat{\Delta\mathbf{l}}$, we find that equation (15) is identical to equation (4).

Applying the update $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$ and iterating, we obtain the undamped *Gauss-Newton (GN) method* as presented in Nocedal and Wright (2006, Chapter 10.3). The method is summarised as Alg. 2 and is recognised to be identical to Alg. 1.

Geometric interpretation

It may be useful to interpret the parameters $\mathbf{x}$ as a vector in the *parameter space* $\mathbb{R}^n$ and the residual $\mathbf{r}(\mathbf{x})$ as a vector in the *observation space* $\mathbb{R}^{\mathrm{m}}$, where n and m are the number of unknowns and observations, respectively. With this interpretation, the residual function $\mathbf{r}(\mathbf{x})$ generates a surface in $\mathbb{R}^{\mathrm{m}}$ and the least squares problem in equation (12) corresponds to finding the point $\boldsymbol{x}_*$ in the parameter space corresponding to the point on the surface $\mathbf{r}(\boldsymbol{x}_*)$ that is closest to the origin, as measured by the $\mathbf{W}$-norm $\|\mathbf{r}\|_{\mathbf{W}}^2 = \mathbf{r}^{\mathrm{T}}\mathbf{W}\mathbf{r}$. The Jacobian $\mathbf{J}(\mathbf{x})$ describes the tangent plane to the surface at $\mathbf{r}(\mathbf{x})$. The solution of the minimisation problem in equation (12) is attained when $\mathbf{J}(\mathbf{x})^{\mathrm{T}}\mathbf{W}\mathbf{r}(\mathbf{x}) = \mathbf{0}$, that is when the residual $\mathbf{r}(\mathbf{x})$ is $\mathbf{W}$-orthogonal to the space spanned by the columns of $\mathbf{J}(\mathbf{x})$ ((a) (B)

FIG.1(b)).

Furthermore, we observe that the ratio

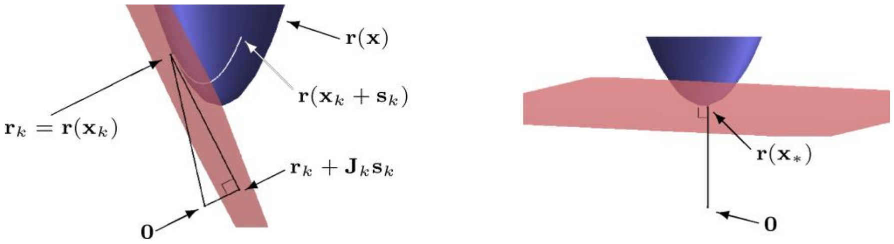
$$\gamma = \frac{\|\mathbf{J}_k\mathbf{s}_k\|_{\mathbf{W}}}{\|\mathbf{r}_k\|_{\mathbf{W}}} = \frac{\left\|\mathbf{J}_k\left(\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{J}_k\right)^{-1}\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{r}_k\right\|_{\mathbf{W}}}{\|\mathbf{r}_k\|_{\mathbf{W}}} \tag{16}$$

is the cosine of the angle (as defined by the $\mathbf{W}$-metric) between the residual $\mathbf{r}_k$ and the tangent plane spanned by $\mathbf{J}_k$ ((A) (b)

FIG.1). Close to the solution, $\mathbf{s}_k$will tend toward zero, whereas the residual $\mathbf{r}_k$ typically remains non-zero. The value of γ can thus be used as a measure of closeness to the optimum.

ALGORITHM 2: *The undamped Gauss-Newton method*

```
1. Start with initial values $\mathbf{x}_0$ of the parameters and $k = 0$.
2. Repeat for $k = 1, 2, \ldots$ “until convergence” or $k > k_{\max}$.
    2.1. Calculate the residual vector $\mathbf{r}_k = \mathbf{r}(\mathbf{x}_k)$ and Jacobian $\mathbf{J}_k = \mathbf{J}(\mathbf{x}_k)$ based on
         the current approximation $\mathbf{x}_k$ of the minimiser.
    2.2. Solve the normal equations (15) for $\mathbf{s}_k$.
    2.3. Update the estimated parameters $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$.
```


$\mathbf{r}(\mathbf{x})$
$\mathbf{r}(\mathbf{x}_k + \mathbf{s}_k)$
$\mathbf{r}_k = \mathbf{r}(\mathbf{x}_k)$
$\mathbf{r}_k + \mathbf{J}_k\mathbf{s}_k$
$\mathbf{0}$
$\mathbf{r}(\mathbf{x}_*)$
$\mathbf{0}$

(a) (b)

FIG.1: Geometric interpretation of a least squares problem with $\mathbf{W} = \mathbf{I}$, corresponding to an unweighted problem. The residual function $\mathbf{r}(\mathbf{x})$ defines a surface in $\mathbb{R}^m$ (here $m = 3$). The solution to the least squares problem is the point on the surface that is closest to the origin $\mathbf{0}$. At the point $\mathbf{r}_k = \mathbf{r}(\mathbf{x}_{\mathrm{k}})$, the surface is locally approximated by the tangent plane spanned by the columns of $\mathbf{J}_k = \mathbf{J}(\boldsymbol{x}_k)$ (a). The point $\mathbf{r}_k + \mathbf{J}_k\mathbf{s}_k$ on the tangent plane closest to the origin is found by the orthogonal projection of the negative residual $\mathbf{J}_k\mathbf{s}_k = \mathbf{J}_k(\mathbf{J}_k^{\mathrm{T}}\mathbf{J}_k)^{-1}\mathbf{J}_k^{\mathrm{T}}(-\mathbf{r}_k)$. The cosine of the acute angle of the triangle is a measure of closeness to the solution. At the optimum $\mathbf{x}_*$, the angle is 90° and the residual $\mathbf{r}(\mathbf{x}_*)$ is orthogonal to the tangent plane (b).

Modifying Gauss-Newton for descent

The update vector $\mathbf{s}_k$ is calculated from a Taylor approximation of the residual function (13) around the current value $\mathbf{x}_k$. The approximation is good in a local neighbourhood around $\mathbf{x}_k$. However, if the updated point $\mathbf{x}_k + \mathbf{s}_k$ is outside the region where the Taylor approximation is good, the new point may be no better than the current point and the method may fail to converge – see Fig. 2.

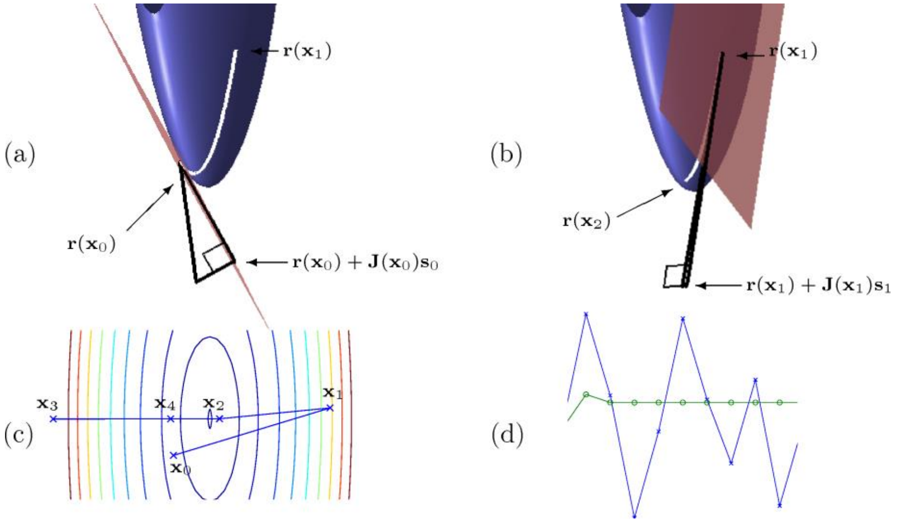

(a)
$\mathbf{r}(\mathbf{x}_1)$
$\mathbf{r}(\mathbf{x}_0)$
$\mathbf{r}(\mathbf{x}_0)+\mathbf{J}(\mathbf{x}_0)\mathbf{s}_0$
(b)
$\mathbf{r}(\mathbf{x}_1)$
$\mathbf{r}(\mathbf{x}_2)$
$\mathbf{r}(\mathbf{x}_1)+\mathbf{J}(\mathbf{x}_1)\mathbf{s}_1$
(c)
$\mathbf{x}_3$
$\mathbf{x}_4$
$\mathbf{x}_2$
$\mathbf{x}_1$
$\mathbf{x}_0$
(d)

FIG. 2: Convergence failure for the undamped GN method. (a) First iteration. The point on the tangent plane closest to the origin is outside the region where the plane approximation is good. The next approximation (b) is on the other side of the minimum. The contour plot (c) in the parameter space shows that the residual increases from $\mathbf{x}_0$ to $\mathbf{x}_1$ and that subsequent iterations oscillate. The component plot (d) shows that first parameter (x) oscillates and the second parameter (o) converges only slowly.

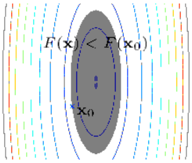


FIG. 3: For a descent method, the next point $\mathbf{x}_{k+1}$must be inside the region defined by the level set $\{\mathbf{x} : F(\mathbf{x}) = F(\mathbf{x}_k)\}$ of the current point $\mathbf{x}_k$.

The GN method may be modified to converge under a large set of circumstances (Nocedal and Wright, 2006). The basic modification is to require that every value is "better" than the previous one. The logical choice is to at least require that the objective function must decrease so that

$$F(\mathbf{x}_{k+1}) < F(\mathbf{x}_k). \tag{17}$$

Algorithms satisfying the descent condition (17) are called *descent methods* – see Fig. 3.

The modifications to make GN a descent method are of two types, *line search* and *trust-region* (Nocedal and Wright, 2006). The line search strategy uses the same linear model in equation (13) that was used to calculate $\mathbf{s}_k$ to decide when a new point is good enough. The trust-region approach works with the quadratic approximation in equation (14) of the objective function for the same purpose. Both methods may be seen as attempts to avoid taking “too long” steps, since when the Jacobian $\mathbf{J}(\mathbf{x}_k)$ is ill-conditioned, the norm of the update $\mathbf{s}_k$can become arbitrarily large. The line search strategy is now considered; the trust-region is discussed in a later section on the Levenberg-Marquardt-Powell method.

Line search

In the line search strategy, the update vector $\mathbf{s}_k$ is calculated by the GN equation in step 2.2 of Alg. 2. However, the update step 2.3 is modified to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k \tag{18}$$

where the scalar α_k is chosen such that the new point $\mathbf{x}_{k+1}$ is better than the current one. In this context, the vector $\mathbf{s}_k$ is known as a *search direction* and the scalar α_k is called the *step length*.

To ensure convergence, the reduction of the objective function cannot be arbitrary small; the descent condition (17) is not enough. One of the simplest additional requirements is that the new point should produce a reduction of the objective function that is at least a fraction μ of the reduction predicted by the linear model used to calculate $\mathbf{s}_k$. This requirement is called the *Armijo condition* (Armijo, 1966; Nocedal and Wright, 2006) and is written as

$$F(\mathbf{x}_k + \alpha_k \mathbf{s}_k) \leq F(\mathbf{x}_k) + \mu \alpha_k \nabla_{\mathbf{x}} F(\mathbf{x}_k)^{\mathrm{T}} \mathbf{s}_k, \tag{19}$$

where $\nabla_{\mathbf{x}} F(\mathbf{x}_k) = \mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{r}_k$. The Armijo constant $0 < \mu < 1$ is fixed whereas the step length α_k is calculated for each iteration.

If α_k is chosen as the first value of the sequence $\{1, ½, ¼, \ldots\}$ that satisfies the Armijo condition, we ensure that we do not take unnecessarily short steps. This technique is known as *backtracking* (Fig. 4). The GN method with Armijo backtracking is presented in algorithms 3 and 4.

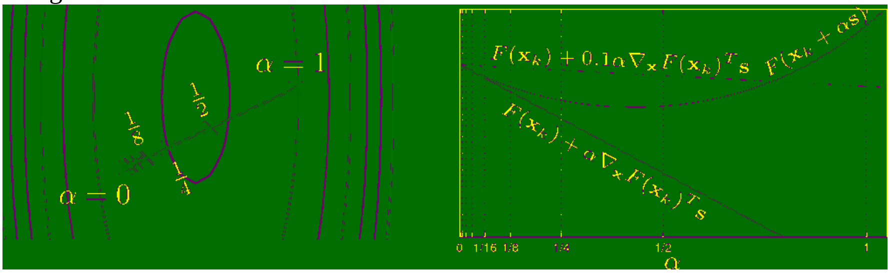


FIG. 4: Armijo line search with backtracking. Left: the line $\mathbf{x}_k + \alpha_k \mathbf{s}_k$ in parameter space for different values of α_k. Right: the objective function value $F(\mathbf{x}_k + \alpha_k \mathbf{s}_k)$ for different values of α_k. The tangent (solid) is a linear predictor of the reduction in objective function value and corresponds to $\mu = 1$. The dashed line corresponds to a fractional reduction of $\mu = 0 \cdot 1$. In the backtracking algorithm 4, the full step $\alpha_k = 1$ is initially tried and discarded since the reduction is not large enough (curve for $\alpha_k = 1$ is above dashed line). Next, $\alpha_k = 1/2$ is tried and accepted since the real reduction is large enough (curve for $\alpha_k = 1/2$ is below dashed line).

ALGORITHM 3: *The Gauss-Newton method with Armijo line search*

```
1. Start with initial values $\mathbf{x}_0$ of the parameters and $k = 0$.
2. Repeat for $k = 1, 2, \ldots$ “until convergence” or $k > k_{\max}$.
    2.1. Calculate the residual vector $\mathbf{r}_k = \mathbf{r}(\mathbf{x}_k)$ and Jacobian $\mathbf{J}_k = \mathbf{J}(\mathbf{x}_k)$ based on
         the current approximation $\mathbf{x}_k$ of the minimiser.
    2.2. Solve the normal equations (15) for $\mathbf{s}_k$.
    2.3. Perform a line search (Alg. 4) to calculate a step length $\alpha_k$ such that the new
         point $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k$ satisfies the Armijo condition.
    2.4. Update the estimated parameters $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k$.
```

ALGORITHM 4: *Armijo line search with backtracking*

```
1. Given an Armijo constant $\mu$ such as $\mu = 0 \cdot 1$.
2. For $\alpha^{(j)} = 2^{-j}, j = 0, 1, \ldots$
    2.1. Calculate a trial point $\mathbf{t}_j = \mathbf{x}_k + \alpha^{(j)} \mathbf{s}_k$.
    2.2. If the trial point $\mathbf{t}_j$ satisfies the Armijo condition (19), return $\alpha^{(j)}$ as the
         current step length $\alpha_k$.
```

Observe that since we first try full steps ($\alpha_k = 1$), the line search method will not interfere with the GN algorithm when the problem is well-behaved. Furthermore, observe that since the Jacobian $\mathbf{J}_k$ and the residual vector $\mathbf{r}_k$ are calculated to find the search direction $\mathbf{s}_k$, the only significant extra calculation required by the line search technique are the objective function values in the trial points.

Levenberg-Marquardt

The line search strategy calculates the search direction $\mathbf{s}_k$ from the unmodified GN equation and restricts the step length by checking how far along $\mathbf{s}_k$ one can go and obtain a reduction of the objective function value. In contrast, the Levenberg-Marquardt (LM) method (Levenberg, 1944; Marquardt, 1963) restricts the length of the search direction $\mathbf{s}_k$ by modifying the GN equation. The modified calculation may be interpreted either as an algebraic modification of the GN equation, a version of ridge regression (Draper and Smith, Jr., 1981, equation 6.7.1), or as a geometric restriction on the length of $\mathbf{s}_k$.

Several popular presentation of the LM method, such as Press et al. (1992) or Hartley and Zisserman (2000), follows the algebraic, "λ-version", algorithm of Marquardt (1963). In this formulation, the update is calculated as the solution of the LM equation

$$\left(\mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{J}_k + \lambda_k \mathbf{I}\right) \mathbf{s}_k = -\mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{r}_k, \tag{20}$$

for a specific value of $\lambda \geq 0$, where $\mathbf{I}$ is the identity matrix. For a given $\mathbf{r}_k$ and $\mathbf{J}_k$, we may write the LM step as an explicit function of λ

$$\mathbf{s}_{LM}(\lambda) = \left(\mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{J}_k + \lambda \mathbf{I}\right)^{-1} \left(-\mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{r}_k\right). \tag{21}$$

Since $\mathbf{s}_{LM}(0) = \mathbf{s}_{GN}$ is the conventional GN step and

$$\mathbf{s}_{LM}(\lambda) \rightarrow -\mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{r}_k / \lambda = -\nabla F(\mathbf{x}_k)/\lambda$$

as $\lambda \rightarrow \infty$, the LM step may be seen as a blend between the GN search direction and a fraction of the negative gradient – see (a) (b)

FIG. 5(a). Another interpretation of the LM damping is as a parameter weighting scheme, where the current estimates are given the weight λ_k. A high λ_k value will bias the next estimate strongly towards the current estimate (suggesting a short step), whereas a small λ_k value will incur a weaker bias (typically suggesting a longer step). For example, for an object point, $\lambda_k = 100$ corresponds to an a priori σ on the coordinates of 0.1 units ($1/\sqrt{100}$) whereas $\lambda_k = 10^{-6}$ corresponds to a σ of 1000 units.

The algebraic LM method adaptively modifies the λ_k value based on the quality of the calculated step. If the step reduces the objective function value, the λ_{k+1}value is decreased, otherwise the λ_{k+1} value is increased and the step is not accepted. The full algorithm is given in Alg. 5.

Observe that there is no simple transition between damped ($\lambda_k > 0$) and undamped ($\lambda_k = 0$) methods. Furthermore, the LM equation (20) must be solved for every new value of λ_k.

Levenberg-Marquardt-Powell (trust-region)

It is possible to interpret the LM method geometrically as a trust-region method (Moré, 1983; Nocedal and Wright, 2006). In this interpretation, the method relies on the same quadratic model (14) as the GN method

$$\Phi_k(\mathbf{s}_k) = \frac{1}{2}\|\mathbf{r}_k + \mathbf{J}_k\mathbf{s}_k\|_{\mathbf{W}}^2 = \frac{1}{2}\mathbf{r}_k^{\mathrm{T}}\mathbf{W}\mathbf{r}_k + \mathbf{s}_k^{\mathrm{T}}\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{r}_k + \frac{1}{2}\mathbf{s}_k^{\mathrm{T}}\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{J}_k\mathbf{s}_k. \quad (22)$$

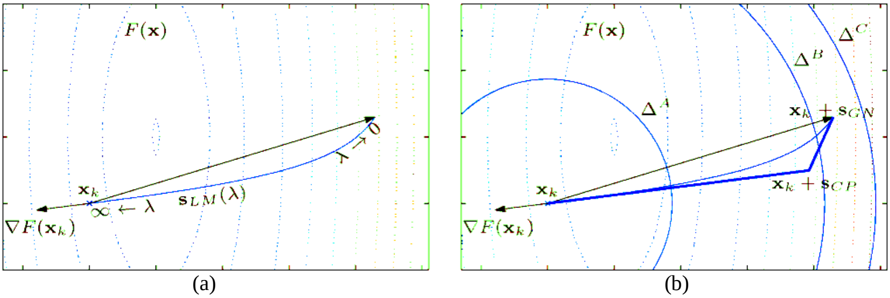


FIG. 5. (a): the LM step varies between the GN search direction ($\mathbf{s}_{LM}(\mathbf{0}) = \mathbf{s}_{GN}$) and a multiple of the negative gradient. (b): the dogleg method finds the intersection between the dogleg path $\mathbf{x}_k \rightarrow \mathbf{x}_k + \mathbf{s}_{CP} \rightarrow \mathbf{x}_k + \mathbf{s}_{GN}$ (thick line) and the circle $\|\mathbf{s}\| = \Delta_k$ or the current value of Δ_k.. If $\|\mathbf{s}_{GN}\| < \Delta_k$, the solution is $\mathbf{s}_{GN}$. The constants Δ^A, Δ^B, Δ^C, are referred to in Alg. 6.

ALGORITHM 5: *The algebraic, "λ-version", Levenberg-Marquardt algorithm*

1. Start with initial values $\mathbf{x}_0$ of the parameters and $k = 0$. Select an initial value of λ_0, such as $\lambda_0 = 10^{-10}\,\mathrm{trace}(\mathbf{J}_0^{\mathrm{T}}\mathbf{W}\mathbf{J}_0)/n$, where $\mathbf{J}_0 = \mathbf{J}(\mathbf{x}_0)$ and n is the number of parameters.
2. Repeat for $k = 1, 2, \ldots$ "until convergence" or $k > k_{\max}$.
    - 2.1. Calculate the residual vector $\mathbf{r}_k = \mathbf{r}(\mathbf{x}_k)$ and Jacobian $\mathbf{J}_k = \mathbf{J}(\mathbf{x}_k)$ unless the last step was rejected.
    - 2.2. Solve the equation system (20) for $\mathbf{s}_k$.
    - 2.3. Calculate a trial point $\mathbf{t}_k = \mathbf{x}_k + \mathbf{s}_k$.
    - 2.4. If $F(\mathbf{t}_k) < F(\mathbf{x}_k)$ (the trial point is better), accept the trial point $\mathbf{x}_{k+1} = \mathbf{t}_k$ and reduce the damping $\lambda_{k+1} = \lambda_k/10$.
    - 2.5. Otherwise (the trial point was worse), stay at the same point $\mathbf{x}_{k+1} = \mathbf{x}_k$ and increase the damping $\lambda_{k+1} = 10\lambda_k$.

The quadratic model is trusted only within a *region of trust* $\|\mathbf{s}_k\| \leq \Delta_k$ around the current approximation $\mathbf{x}_k$. Thus, at each iteration we consider the constrained sub-problem

$$\min_{\mathbf{s}_k} \frac{1}{2}\mathbf{r}_k^{\mathrm{T}}\mathbf{W}\mathbf{r}_k + \mathbf{s}_k^{\mathrm{T}}\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{r}_k + \frac{1}{2}\mathbf{s}_k^{\mathrm{T}}\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{J}_k\mathbf{s}_k, \quad \text{subject to } \|\mathbf{s}_k\| \leq \Delta_k. \tag{23}$$

Mathematically, the solution of problem (23) is given by equation (20) for some value of the Lagrange multiplier $\lambda_k \geq 0$. Indeed, equation (20) may be seen as a way to solve problem (23).

Powell dogleg

The method of Powell (1970) may be used to solve problem (23) without any knowledge of the λ_k value. The method is called *dogleg* due to the shape of the piecewise linear path used to approximate the $\mathbf{s}_{LM}(\lambda)$ curve. The dogleg path goes from the current point, via the *Cauchy point* $\mathbf{s}_{CP}$, to the GN search direction $\mathbf{s}_{GN}$ – see (a)
(b)

FIG. 5(b). The Cauchy point is defined as

$$\mathbf{s}_{CP} = \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k}{\mathbf{g}_k^{\mathrm{T}}\mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{J}_k\mathbf{g}_k}(-\mathbf{g}_k), \tag{24}$$

where $\mathbf{g}_k = \nabla F(\mathbf{x}_k) = \mathbf{J}_k^{\mathrm{T}}\mathbf{W}\mathbf{r}_k$ is the minimiser of the quadratic model $\Phi_k(\mathbf{s}_k)$ in the direction of the negative gradient. Given the current value of Δ_k, the dogleg point is found as the intersection of the dogleg path with the circle $\|\mathbf{s}_k\| = \Delta_k$. If $\|\mathbf{s}_{GN}\| < \Delta_k$, the dogleg point is chosen as $\mathbf{s}_{GN}$.

During the iterations, the trust region size Δ_k is adaptively modified based on how well the quadratic model predicts the reduction of the objective function. The *gain ratio* ρ_k between the actual and predicted reductions is defined as

$$\rho_k = \frac{F(\mathbf{x}_k) - F(\mathbf{x}_k + \mathbf{s}_k)}{\Phi_k(\mathbf{0}) - \Phi_k(\mathbf{s}_k)} \tag{25}$$

If ρ_k is high enough, the step is accepted and the region size Δ_k is increased. If ρ_k is too low, the step is discarded and the region size Δ_k is decreased. The full algorithm is given as Alg. 6.

ALGORITHM 6: *The geometric Levenberg-Marquardt-Powell algorithm*

```
1. Start with initial values $\mathbf{x}_0$ of the parameters and $k = 0$. Select an initial value of $\Delta_0$, such as $\Delta_0 = \|\mathbf{x}_0\|$.
2. Repeat for $k = 1, 2, \ldots$ “until convergence” or $k > k_{\max}$.
   2.1. If the last step was accepted:
      2.1.1. Calculate the residual vector $\mathbf{r}_k = \mathbf{r}(\mathbf{x}_k)$ and Jacobian $\mathbf{J}_k = \mathbf{J}(\mathbf{x}_k)$.
      2.1.2. Solve the equation system (15) for $\mathbf{s}_{GN}$.
      2.1.3. Calculate the Cauchy point $\mathbf{s}_{CP}$ (eq. (24)).
   2.2. Solve the constrained problem (23) for a dogleg point $\mathbf{s}_k$:
      2.2.1. If $\|\mathbf{s}_{GN}\| \leq \Delta_k, \mathbf{s}_k = \mathbf{s}_{GN}$. ($\Delta_k = \Delta^C$ in (a)
             (b)
      2.2.2. FIG. 5).
      2.2.3. Otherwise, if $\|\mathbf{s}_{CP}\| \geq \Delta_k, \mathbf{s}_k = \Delta_k\mathbf{s}_{CP}/\|\mathbf{s}_{CP}\|$. ($\Delta_k = \Delta^A$ in (a)
             (b)
```

2.2.4. FIG. 5).

2.2.5. Otherwise, find the intersection of the line $\mathbf{x}_k + \mathbf{s}_{CP} \rightarrow \mathbf{x}_k + \mathbf{s}_{GN}$ and a circle with radius $||\Delta\_k||$. ($\Delta_k = \Delta^A$ in (a) (b)

2.2.6. FIG. 5).

2.3. Calculate a trial point $\mathbf{t}_k = \mathbf{x}_k + \mathbf{s}_k$.

2.4. Calculate the gain ratio ρ_k (equation (25)).

2.5. If $\rho_k < 0 \cdot 25$ (the prediction is bad), discard the step $\mathbf{x}_{k+1} = \mathbf{x}_k$ and decrease the trust region size to $\Delta_{k+1} = \Delta_k/2$.

2.6. Otherwise if $\rho_k < 0 \cdot 75$ (the prediction is fair), accept the step $\mathbf{x}_{k+1} = \mathbf{t}_k$ but maintain the trust region size so $\Delta_{k+1} = \Delta_k$.

2.7. Otherwise (the prediction is good), accept the step $\mathbf{x}_{k+1} = \mathbf{t}_k$ and increase the trust region size to $\Delta_{k+1} = 2\Delta_k$.

Compared to the algebraic LM method, the geometric method handles the transition to an undamped solution transparently. Furthermore, the additional cost for calculating dogleg points for different values of Δ_k is the calculation of a square root.

Problem-specific damping

In algorithms 3 to 6, a trial point $\mathbf{t}_k$ is calculated and tested at each iteration. If $\mathbf{t}_k$ does not represents a good enough improvement over the current point, it is rejected and another point is tried according to an algorithm-specific scheme. The quality of $\mathbf{t}_k$ is judged by the reduction of the objective function value.

This process is valid for any non-linear least squares problem. However, it is possible to add a problem-specific *veto* condition to disqualify “illegal” trial points. For the BA problem, an illegal trial point $\mathbf{t}_k$ may, for instance, violate the chirality constraint (each object point should be in front of each camera in which the object point was measured). The veto condition would be added to steps 2.2, 2.4, and 2.5 of algorithms 4, 5 and 6, respectively. Thus, with the veto condition, a trial point will be accepted only if the objective function value is reduced *and* the trial point satisfies the veto condition. Importantly, for the veto addition to work, the initial values must satisfy the veto condition.

EXPERIMENTS

A perturbation study was performed to investigate the pull-in radius of the presented algorithms. Four BA algorithms were implemented in Matlab and applied to a terrestrial BA network. The implementations used Matlab’s sparse matrix library where sparse linear equation systems are solved via sparse Cholesky factorisation by the CHOLMOD algorithm (Davis, 2009).

Bundle algorithms

The implemented LSA algorithms were:

GM – the classical Gauss-Markov algorithm (Alg. 1) with no damping.

GNA – Gauss-Newton with Armijo line search (Alg. 3).
LM – the original Levenberg-Marquardt algorithm (Alg. 5).
LMP – Levenberg-Marquardt with Powell dogleg (Alg. 6).

Each damped algorithm was available with the standard and the chirality based veto damping. The datum problem was handled by fixing the first camera and one coordinate of another camera, i.e. dependent relative orientation (Mikhail et al., 2001). Furthermore, a camera model using the ω-ϕ-κ Euler angle convention (Förstner and Wrobel, 2004) and parameters K_1, K_2, K_3, P_1, P_2 of the Brown (1971) lens distortion model was implemented. The ω-ϕ-κ convention is used by the software in many programs and is valid as long as the middle angle is not close to 90°. In this paper, a combination of the camera model and an LSA algorithm is called a BA algorithm. The algorithms assume fixed, known IO parameters and do not perform any outlier detection during the least squares iterations.

The values of the algorithm-specific constants were chosen as follows. The Armijo constant $\mu = 0 \cdot 1$ used by GNA was chosen as a sound value based on the experience of the first author. For the LM algorithm, λ values below an arbitrary cutoff $\lambda_c = 10^{-10}\text{trace}(\mathbf{J}_0^{\mathrm{T}}\mathbf{W}\mathbf{J}_0)/n$ were considered to be zero. The initial λ value was chosen as $\lambda_0 = \lambda_c$ to enable the LM algorithm to quickly switch to the undamped mode for well-behaved problems. The choice of $\Delta_0 = \|\mathbf{x}_0\|$ for the LMP method is based on the assumption that the initial values are correct to within one order of magnitude. The convergence test for all algorithms was that the closeness ratio γ of equation (16) was below $\gamma_c = 10^{-3}$, again based on the experience of the first author.

Network

The bundle network used in the experiments was from a 60-image dataset of the Arco di Constantino in Rome, Italy, acquired by a Canon EOS 5D Mark II camera with a 24 mm lens (Fig. 6). The images were processed by the SmartMatch feature in Photomodeler 2011 (www.photomodeler.com) with the *Ordered Photos Surround* option. The SmartMatch feature detects natural features in the images and matches them between images. A total of 26 321 object points with an average ray count of 3·4 was obtained. The average ray intersection angle was 23°, with the smallest being 1·2°. All observations were given unit weight. The resulting $\widehat{\sigma_0} = 0 \cdot 61$ thus corresponds to an estimated coordinate measurement σ of 0·61 pixels. After processing, the calibrated camera parameters and measured x, y coordinates were exported to Matlab.

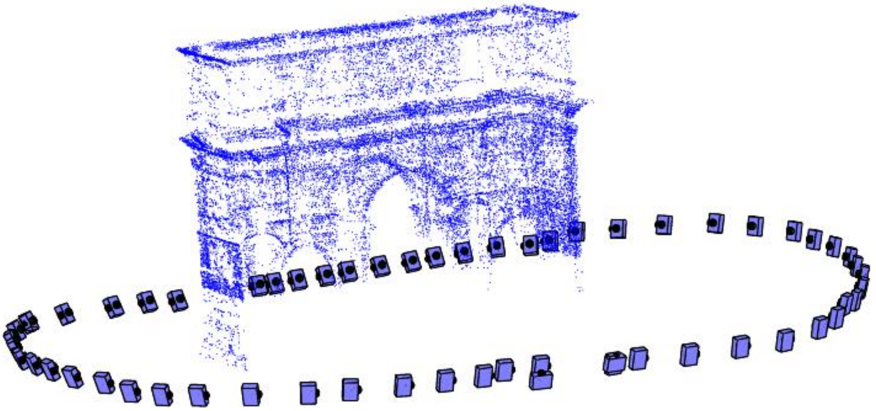


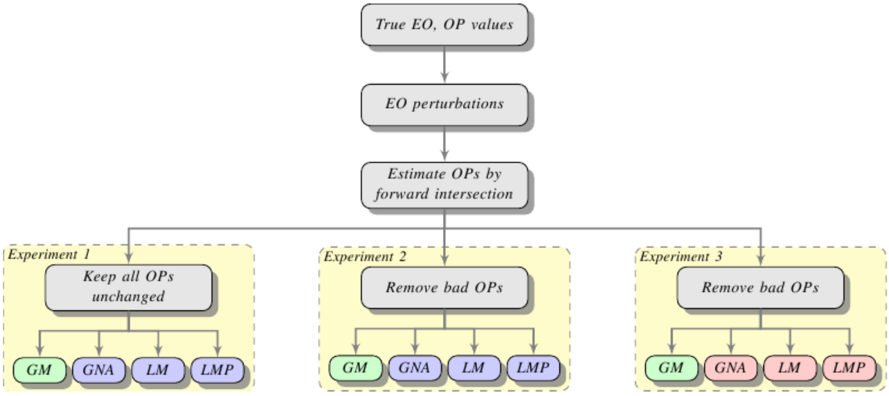
FIG. 6. A 60-image network of the *Arco di Constantino* in Rome, Italy, acquired by a Canon EOS 5D Mark II camera was used in the experiments. This figure shows the ground truth EO and OP parameters.

Perturbations

To determine a reasonable limit on the perturbation size to be investigated, the Nistér (2004) five-point algorithm was applied to each sequential image pair. For each image pair, the Euler angles of the right image relative to the left image, as calculated by the Nistér algorithm, was subtracted from the corresponding optimal values (see Appendix A for details). The RMS of all angle differences and networks was 0·3°. The largest recorded difference was 2·3°. To stress test the algorithms, an upper limit of 3° was chosen for the experiments. For the perturbation of the camera position, a limit of 4% of the object size $D = 22$m was chosen.

Setup

The optimal EO and OP parameters were estimated by one of the bundle algorithms based on the best available initial values. Consensus on the optimal values was verified by the other bundle algorithms. The EO and OP estimates were thereafter considered as “ground truth”.



Based on the ground truth values, three experiments were constructed (see Fig. 7). In all experiments, the initial EO values were taken as random perturbations of varying sizes of the ground truth values. The initial OP values were estimated by forward intersection based on image measurements and the initial EO values (see Appendix B).

Estimating the initial OP values based on uncertain EO parameters will sometimes produce "bad" OPs that appear behind one or more of the cameras in which the OPs are visible. The experiments treated this situation differently. In Experiment 1, all initial OP values were left unchanged and given to GM and each of the standard versions of the dam BA algo of the ith conv ns, and (iii) the execution time, was recorded.

FIG. 7. Overview of the experiments. All experiments started by perturbing the EO values followed by forward intersection to estimate the OP values. The experiments differ in two respects: (i) whether the standard formulation (used in Experiments 1-2) or veto formulation (Experiment 3) of the damped algorithms was used; and (ii) in the handling of "bad" OPs, which are initial OPs that appear behind any cameras. The OPs were either left unchanged (Experiment 1) or removed (Experiments 2-3).

Each algorithm was tested on angle and position perturbations of β = 0, 0·5, $\cdots$, 3 degrees and d = 0, 1, $\cdots$, 4 per cent, respectively. A perturbation level of β degrees and d percent corresponds to a maximum perturbation of $\pm\beta°$ on each camera angle and $\pm dD$ on each camera coordinate, where $D = 22\text{m}$ is the object size. For each experiment block, in other words the combination of perturbation sizes, a total of n = 250 perturbations of the optimal values were used as initial values. Timings were performed on a Lenovo Thinkpad T420 with Intel Core i7-2620 at 2.7GHz, running Matlab 7.13.0 (R2011b) under 64-bit Linux 2.6.38.

RESULTS

The convergence results are presented in Table I. The results are given for each method and network with the standard and veto formulations. Each table entry is the percentage of n = 250 simulations that converged to the true optimal values for a given perturbation level. Convergence percentages of at least 99% are called *acceptable*. Values below 99% and above 75% are called *intermediate*.

Table I: Convergence results for the three experiments. The percentage of n = 250 runs that achieved convergence for varying levels of angular and translational perturbation are given. Only values above 75% are presented. Values 75% to 99% are in grey to improve readability. Some repeated rows and columns with no information have been omitted.

(a) Experiment 1. Bad OPs unchanged, standard damped algorithms

GM	*% of object size*					GNA	*% of object size*					LM	*% of object size*					LMP	*% of object size*				
	0	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*
0·0°	100	99	-	-	-	0·0°	100	99	-	-	-	0·0°	100	99	-	-	-	0·0°	100	99	-	-	-
0·5°	88	85	-	-	-	0·5°	92	87	-	-	-	0·5°	91	88	-	-	-	0·5°	92	87	-	-	-
1·0°	-	-	-	-	-	1·0°	-	-	-	-	-	1·0°	-	-	-	-	-	1·0°	-	-	-	-	-

(b) Experiment 2. Bad OPs removed, standard damped algorithms

GM	*% of object size*	GNA	*% of object size*	LM	*% of object size*	LMP	*% of object size*

	0	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*
0·0°	100	99	95	-	-	0·0°	100	100	98	-	-	0·0°	100	100	97	-	-	0·0°	100	100	98	-	-
0·5°	88	88	-	-	-	0·5°	92	90	82	-	-	0·5°	91	91	77	-	-	0·5°	92	90	82	-	-
1·0°	-	-	-	-	-	1·0°	-	-	-	-	-	1·0°	-	-	-	-	-	1·0°	-	-	-	-	-

(c) Experiment 3. Bad OPs removed, veto damped algorithms

GM	*% of object size*					**GNA**	*% of object size*					**LM**	*% of object size*					**LMP**	*% of object size*				
	0	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*		*0*	*1*	*2*	*3*	*4*
0.0°	100	100	95	-	-	0.0°	100	100	100	94	80	0.0°	100	100	100	91	-	0.0°	100	100	100	94	76
0.5°	88	88	-	-	-	0.5°	100	100	100	93	78	0.5°	100	100	100	88	-	0.5°	100	100	100	92	-
1.0°	-	-	-	-	-	1.0°	100	100	99	94	-	1.0°	92	92	87	-	-	1.0°	100	100	100	90	-
1.5°	-	-	-	-	-	1.5°	100	99	98	88	-	1.5°	-	-	-	-	-	1.5°	100	100	98	90	-
2.0°	-	-	-	-	-	2.0°	99	99	96	88	-	2.0°	-	-	-	-	-	2.0°	99	100	96	86	-
2.5°	-	-	-	-	-	2.5°	94	94	89	81	-	2.5°	-	-	-	-	-	2.5°	99	98	93	79	-
3.0°	-	-	-	-	-	3.0°	89	89	87	-	-	3.0°	-	-	-	-	-	3.0°	96	98	92	77	-

In Experiment 1 (Table I(a)), where all OPs were kept, all algorithms had the same pull-in region, as defined by the acceptable results. The intermediate results were slightly higher for the damped algorithms. In Experiment 2 (Table I(b)), the removal of the bad OPs did not change the pull-in region. However, the number and values of the intermediate results increased for all algorithms. In contrast, in Experiment 3 (Table I(c)), the damped veto algorithms – especially GNA and LMP – increased the pull-in region markedly. The largest change was seen with respect to angular perturbations. In all experiments, the number and/or the values of the intermediate results were higher for the damped algorithms – especially GNA and LMP – than for the classical algorithm.

The timing and iteration count between the algorithms was compared for the 250 runs of the 0·0°/1% block where all algorithms almost always worked. On average, the GM, GNA, and LMP methods required 3·3–3·5 iterations and 5·4–5·6 seconds to converge. The LM method required about one further iteration and 15–20% longer execution time. The veto damping added about 3% in execution time and had insignificant effect on the iteration count.

DISCUSSION

Bundle adjustment (BA) has been used in photogrammetry since the 1960s (Schmid, 1956; Brown, 1958; 1976; Granshaw, 1980; Luhmann et al., 2006). Several important improvements of the overall BA algorithm have been presented, including methods to detect outliers (Baarda, 1968), the use of GPS as additional EO observations (for example Ackermann, 1994) to stabilise the method and reduce the dependency on ground control points, and automatic relative orientation procedures (Läbe et al., 2008) to establish initial values. Several papers have focused on improving the speed of a single iteration of the least squares adjustment (LSA) by utilising the sparsity pattern of the normal matrix (for example Brown, 1976; Kraus, 1993). However, the LSA iteration sequence has largely remained unchanged.

In a review paper by Triggs et al. (2000), multiple damped (called *step control*) iterative BA algorithms are discussed, including Levenberg-Marquardt and line-search. The textbook by Hartley and Zisserman (2000) suggests the LM algorithm as the main estimation algorithm; furthermore, the authors argue (pages 94–95) that the LM method can and should be used on over-parameterised problems to reduce the risk of becoming trapped in a non-global minimum. This use of the LM method has been criticised by several authors including Gruen and Akca (2005), where the addition of the λ parameter to the normal equations (20) was seen as purely a numerical way to avoid singular normal matrices, as would be the case with an over-parameterised functional model. Indeed, if the LM algorithm converges with a non-zero λ, the a posteriori estimates of equations (5) to (8) will be biased. The size of the bias will depend on the problem and on the implementation of the algorithm. However, on a problem which is not over-parameterised, and if the LM method converges with a final value of $\lambda = 0$, the a posteriori estimates of equations (5) to (8) are again unbiased.

Within the photogrammetric community, Jianchar and Chern (2001) compared the undamped GM method (called *Newton-Gauss*) with the LM method on the spatial resection problem. On the BA problem, Börlin et al. (2004) suggested the GNA algorithm (called *Alg. 1D*) and compared it to the GM method (called *Alg. 1U*) on a problem with two cameras. The conclusion was that the GNA method reduced the number of convergence failures compared to GM at an insignificant extra computational cost. Later, Lourakis and Argyros (2005) applied the LM and LMP (called *DL*) algorithms on the BA problem and concluded that LMP was faster than LM. Despite this, the LM version of the Levenberg-Marquardt method was chosen for their publicly-available SBA software package (Lourakis and Argyros, 2009).

The focus of this paper has been to study the convergence properties of damped and undamped versions of the iterative adjustment part of the BA algorithm. The intention has been to solve the *same* estimation problem using the *same* data and initial values and producing the *same* result. Furthermore, the number of observations was kept constant during the iterations. The introduction of a priori observations of some parameters, although a generally useful technique, does modify the result and has therefore not been studied. Furthermore, the focus has strictly been on LSA algorithms, so M-estimators (Triggs et al., 2000) or convex optimisation methods (Kahl and Hartley, 2008) have not been studied.

In previous BA papers (Börlin et al., 2004; Lourakis and Argyros, 2005), the results were based on iterations from a single starting point. In this paper, a perturbation study was performed to investigate the pull-in radius of the GM, GNA, LM, and LMP algorithms on a terrestrial BA problem with 60 images. The perturbations were applied to the EO parameters, with the OP parameters subsequently determined by forward intersection. The size of the perturbations varied from 0 to 3 degrees on each Euler angle and from 0 to 4 per cent of the object size on each camera coordinate. The angular limit was chosen to cover the expected real angular error, which was approximated to be around 2° on the same network. The positional limit was chosen to show differences between the algorithms. In addition to the standard non-linear optimisation damping techniques, this paper introduces a novel “veto” damping based on the chirality condition: that each OP should be in front of each camera in which it was measured during every iteration.

The results show that in Experiments 1 and 2, the damped algorithms had no greater pull-in region than the classical algorithm. However, outside the pull-in region, the damped algorithms did show a less marked drop-off in performance. This is consistent with theory presented in, for instance, Nocedal and Wright (2006) and the results by Jianchar and Chern (2001) and Börlin et al. (2004). Compared to Experiment 1, the removal of bad OPs in Experiment 2 provided only marginal improvement. However, when the same data was given to the veto algorithms in Experiment 3 the pull-in region, especially for the GNA and LMP algorithms, increased dramatically.

Compared to GM, an insignificant extra cost in time and/or iterations was seen for GNA and LMP. The LM method required about one additional iteration and about 15–20% longer execution time. The relative performance of LM and LMP is consistent with the results by Lourakis and Argyros (2005). The additional cost of using veto iterations was around 3% in execution time.

Outlier detection during the least squares iterations was not part of the current investigation. This was partly due to the fact that it is non-trivial to modify the damped algorithms to add or remove observations during the iterations and still remain descent methods. Instead, outlier detection was used at the initial value stage in Experiments 2 and 3; otherwise it was deferred to after convergence and hence is outside the scope of this paper.

CONCLUSION

This paper has shown that damped algorithms are less sensitive to errors in the initial EO and OP values compared to the classical algorithm. The possibility that undetected outliers remain in the data used cannot be ruled out, and could perhaps explain the small size of the convergence region in experiments 1 and 2. However, the presence of outliers in the data would not change the main conclusion of the paper, namely that the damped algorithms are less sensitive than classical methods.

From a photogrammetric point of view it could be argued that the proper solution to convergence problems would be to add more information to the bundle network. The authors agree that this should always be considered. The statistical quality of the final result is given by the covariance analysis at the minimum, not the method used to arrive at the minimum. Thus, damping is no miracle cure that can improve the quality of a poorly designed network. However, these results suggest that damped methods can in many cases provide a solution where an undamped method fails.

In conclusion, the authors believe that damping — preferably line-search or Levenberg-Marquardt-Powell — should be available in any software package for least squares adjustment. Future work includes the extension of the damped techniques to problems with functional constraints.

REFERENCES

ACKERMANN, F., 1994. Practical experience with GPS-supported aerial triangulation. *Photogrammetric Record,* 16(84): 860-874.

ARMIJO, L., 1966. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics,* 16 (1): 1–3.

BAARDA, W., 1968. A testing procedure for use in geodetic network. *Netherlands Geodetic Commission, Publications on Geodesy,* 2(5). Delft, The Netherlands. 97 pages.

BÖRLIN, N., GRUSSENMEYER, P., ERIKSSON, J. and LINDSTRÖM, P., 2004. Pros and cons of constrained and unconstrained formulation of the bundle adjustment problem. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences,* 35 (B3): 589–594.

BROWN, D. C., 1958. A solution to the general problem of multiple station analytical stereotriangulation. *Ballistic Research Laboratories, Technical Report 43.* Aberdeen Proving Ground, Maryland, USA.

BROWN, D. C., 1971. Close-range camera calibration. *Photogrammetric Engineering,* 37 (8): 855–866.

BROWN, D. C., 1976. The bundle adjustment: progress and prospects. *International Archives of Photogrammetry,* 21(3): 33 pages.

DAVIS, T. A., 2009. User guide to CHOLMOD: a sparse Cholesky factorization and modification package. http://www.cise.ufl.edu/research/sparse/cholmod/CHOLMOD/Doc/UserGuide.pdf.

DRAPER, N. R. and SMITH, JR., H., 1981. *Applied Regression Analysis*, 2nd Edition, John Wiley, New York, USA. 709 pages.

FÖRSTNER, W. and WROBEL, B., 2004. Mathematical concepts in photogrammetry. In: *Manual of Photogrammetry* (Eds. J. McGlone and E. Mikhail), American Society of Photogrammetry and Remote Sensing, 5th Edition. Bethesda, Maryland, USA. 1200 pages: 15-181.

GRANSHAW, S. I., 1980. Bundle adjustment methods in engineering photogrammetry. *Photogrammetric Record*, 10(56): 181-207.

GRUEN, A. and AKCA, D., 2005. Least squares 3d surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing,* 59 (3): 151– 174.

HARTLEY, R. I. and ZISSERMAN, A., 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK. 607 pages.

JIANCHAR, Y. and CHERN, C. T., 2001. Comparison of Newton-Gauss with Levenberg-Marquardt algorithm for space resection. *Proceedings of the 22nd Asian Conference on Remote Sensing,* Singapore, 1: 256–261.

KAHL, F. and HARTLEY, R., 2008. Multiple-view geometry under the L_∞-norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 30(9):1603–1617.

KRAUS, K., 1993. *Photogrammetry, Fundamentals and Standard Processes.* Volume 1. Dummler Verlag, Bonn, Germany. 389 pages.

LÄBE, T., DICKSCHEID, T. and FÖRSTNER, W., 2008. On the quality of automatic relative orientation procedures. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* 37(B3b): 37-42.

LEVENBERG, K., 1944. A method for the solution of certain nonlinear problems in least squares. *Quarterly of Applied Mathematics,* 2: 164–168.

LOURAKIS, M. I. A. and ARGYROS, A. A., 2005. Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? *Proceedings of IEEE International Conference on Computer Vision,* Beijing, China, 2: 1526–1531.

LOURAKIS, M. I. A. and ARGYROS, A. A., 2009. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1):1–30.

LUHMANN, T., ROBSON, S., KYLE, S. and HARLEY, I., 2006. *Close Range Photogrammetry: Principles, Methods and Applications*. Whittles, Scotland. 510 pages.

MARQUARDT, D. W., 1963. An algorithm for least squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics,* 11: 431-441.

MCGLONE, C., MIKHAIL, E. and BETHEL, J. (Eds.), 2004. *Manual of Photogrammetry,* Fifth Edition. American Society for Photogrammetry and Remote Sensing. Bethesda, Maryland, USA. 1200 pages.

MIKHAIL, E., BETHEL, J. and MCGLONE, C. (Eds.), 2001. *Introduction to Modern Photogrammetry*. Wiley, New York, USA. 496 pages.

MORÉ, J. J., 1983. Recent developments in algorithms and software for trust region methods. In: *Mathematical Programming — The State of the Art.* (Eds. A. Bachem, M. Grötschel and B. Korte), Springer, Berlin, Germany. 655 pages: 258–287.

NISTÉR, D., 2004. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*: 26(6): 756-770.

NOCEDAL, J. and WRIGHT, S. J., 2006. *Numerical Optimization*. Second Edition. Springer, Berlin, Germany. 664 pages.

POWELL, M. J. D., 1970. A hybrid method for nonlinear equations. *Numerical Methods for Nonlinear Algebraic Equations* (Ed. P. Rabinowitz). Gordon and Breach Science, London, UK. 200 pages: 87–144.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. and FLANNERY, B. P., 1992. *Numerical Recepies in C*, Second Edition. Cambridge University Press, Cambridge, UK. 256 pages.

SCHMID, H. H., 1956. An analytical treatment of the problem of triangulation by stereophotogrammetry. *Photogrammetria* 13: 67–77.

STRANG, G. and BORRE, K., 1997. *Linear Algebra, Geodesy, and GPS*. Wellesley-Cambridge Press, Wellesley, Massachusetts, USA. 624 pages.

TRIGGS, B., MCLAUCHLAN, P., HARTLEY, R. and FITZGIBBON, A., 2000. Bundle adjustment – a modern synthesis. In*: Vision Algorithms: Theory and Practice, Proceedings of the International Workshop on Vision Algorithms.* Lecture Notes in Computer Science 1883: 298–372.

APPENDIX A: ESTIMATION OF REAL PERTURBATION

1. Assume the N images are sorted from left to right.
2. Run the bundle adjustment on the complete network to obtain the real external orientation of each image.
3. For each image $l = 1, 2, \cdots, N$, in the network:
    - 3.1. Align the network result of step 2 such that the orientation of image l coincides with the global coordinate system.
    - 3.2. Extract the Euler angles of camera $r = l + 1$ into the vector $\mathbf{v}_l^*$.
4. For each image $l = 1, 2, \cdots, N$:
    - 4.1. Determine the set $\mathcal{O}$ of OPs visible in images l and $r = l + 1$.
    - 4.2. Determine the relative orientation of image r with respect to image l using the 5-point algorithm of Nistér (2004) on the set $\mathcal{O}$.
    - 4.3. Extract the Euler angles of image r into the vector $\hat{\mathbf{v}}_l$.
5. Compute the RMS and maximum of the angle differences:

$$\{\Delta\mathbf{v}_l = \hat{\mathbf{v}}_l - \mathbf{v}_l^*\}, l = 1, 2, \dots, N.$$

The image network is considered cyclic, so image 1 is considered to be to the right of image N.

APPENDIX B: CALCULATION OF INITIAL VALUES

Given optimal values $\{\mathbf{c}_i^*\}$ and $\{\mathbf{v}_i^*\}$ of the position and angles, respectively, for each camera and a maximum perturbations size d and β, respectively, the initial EO and OP values are calculated by the following algorithm:

1. For each image i:
    - 1.1. Initialise the image position and orientation parameters as perturbations of the optimal values:

$$\mathbf{c}_i = \mathbf{c}_i^* + d(2\mathbf{p} - 1),$$
$$\mathbf{v}_i = \mathbf{v}_i^* + \beta(2\mathbf{q} - 1),$$

where $\mathbf{p}$ and $\mathbf{q}$ are vectors of independent uniform random numbers between 0 and 1.

2. Estimate the position $\mathbf{p}_j$ of each object point j by forward intersection.
3. (For Experiments 2 and 3 only.) For each object point j:
    - 3.1. Determine the set $\mathcal{S}_j$ of images the point $\mathbf{p}_j$ was measured in.
    - 3.2. If $\mathbf{p}_j$ is behind any camera in $\mathcal{S}_j$:
        - 3.2.1. Remove $\mathbf{p}_j$.