Robotica (2016) volume 34, pp. 703–722. © Cambridge University Press 2014
doi:10.1017/S0263574714001829

Robust L_∞ convex optimisation for monocular visual odometry trajectory estimation

Mohammed Boulekchour*, Nabil Aouf and Mark Richardson

Department of Informatics and Systems Engineering, Cranfield University, Shrivenham SN6 8LA, United Kingdom
E-mails: n.aouf@cranfield.ac.uk, m.a.richardson@cranfield.ac.uk

(Accepted June 11, 2014. First published online: July 9, 2014)

SUMMARY
The most important applications of many computer vision systems are based on robust features extraction, matching and tracking. Due to their extraction techniques, image features locations accuracy is heavily dependent on the variation in intensity within their neighbourhoods, from which their uncertainties are estimated. In the present work, a robust L_∞ optimisation solution for monocular motion estimation systems has been presented. The uncertainty estimation techniques based on SIFT derivative approach and its propagation through the eight-point algorithm, singular value decomposition SVD and the triangulation algorithm have proved an improvement to the global motion estimation. Using monocular systems makes the motion estimation challenging due to the absolute scale ambiguity caused by projective effects. For this, we propose robust tools to estimate both the trajectory of a moving object and the unknown absolute scale ratio between consecutive image pairs. Experimental evaluations showed that robust convex optimisation with the L_∞ norm under uncertain data and the Robust Least Squares clearly outperform classical methods based on Least Squares and Levenberg-Marquardt algorithms.

KEYWORDS: Robust optimisation; Uncertainty propagation; L_∞ Norm; Convex optimisation; Monocular visual odometry; SOCP.

1. Introduction

In the recent years a need to deal with uncertain data has become crucial especially when real life applications are involved. In these circumstances, robust optimisation aims to recover an optimal solution whose feasibility must guaranteed for any realisation of the uncertain data.[1] Robust optimisation for which the data are not specified exactly has come to explicitly incorporate uncertainty to protect the decision-maker against parameters ambiguity and stochastic uncertainties.[2]

Visual Odometry (VO) types of approaches have been widely studied in the last decade as a possible solution for autonomous navigation systems. Optimisation techniques, such as bundle adjustment are used to deliver trajectory estimates. Typically, the camera pose is estimated from the available corresponding points between two views and the camera calibration parameters. Commonly, essential matrix estimation is performed through the eight-point algorithm which is then used to recover the camera pose by solving linear least-squares problems via Singular Value Decomposition SVD.[3] A successful technique in ref. [4] for real time motion estimation of a single camera or stereo rig is presented. While stereo visual odometry takes advantage of the known baseline to directly remove any scale ambiguity, it suffers from huge lack of accuracy when the distance to the 3D landmarks (features) is much larger than the baseline. Hence, monocular VO becomes more practical to be adopted as the former reduces to the monocular case.[5] On the other hand, monocular algorithms suffer from the scale ambiguity. Esteban *et al.* in ref. [6] presented a successful monocular visual

* Corresponding author. E-mail: m.boulekchour@cranfield.ac.uk

odometry algorithm relying on a linear computation of the scale ratio between frame pairs using Least Squares algorithm. Real world information such as cameras height to the ground plane and position with respect to the axis of the vehicle, could be used as well to estimate the global scale factor to retrieve absolute motion from monocular camera.[17]

Bundle adjustment algorithm has been widely used in motion estimation for both stereo and monocular systems. It is an optimisation technique for refining a visual reconstruction jointly with optimal camera pose estimates by minimising the re-projection error. This optimisation problem is usually formulated as a non-linear least squares problem, where the error is the squared L_2 norm of the difference between the observed feature location and the projection of the corresponding 3D point on the image plane of the camera. Levenberg-Marquardt (LM) algorithm is the most popular algorithm for solving non-linear least squares problems, and the algorithm of choice for bundle adjustment. However, the main issue with these methods, and notwithstanding of dependency on good initialisation, is related to the high probably of converging to a local minimum or even to an infeasible solution.

As a powerful alternative, convex optimisation (CVX) offers the possibility of getting around issues when dealing with these non-linear minimisation problems.[7,8] A projective bundle adjustment algorithm using L_∞ norm is proposed by Mitra and Chellappa[9] based on minimising the L_∞ norm of re-projection error where the problem is divided into two successive tasks by fixing the parameters of one sub-problem while optimising the remaining sub-problem using convex optimisation. The first sub-problem is linked to recovering the camera parameters while keeping the structure parameters fixed. The second one is linked to fixing the camera parameters and estimating structure parameters. In ref. [10], a stereo visual odometry approach determining the essential matrix by minimizing the algebraic error through a convex optimisation is presented.

Dealing with uncertain data and more specifically in our visual uncertain data has become crucial. Robust optimisation is able to recover an optimal solution that guarantees its feasibility for any realisation of the uncertain data.[1] Indeed, robust optimisation for which data are not precisely specified can explicitly incorporate uncertainty.[2] Knowing that image-based measurements are subject to deterministic perturbations, more studies have started to focus on how parameters estimation using these measurements might be improved if additional information characterising the uncertainty of the data is available.[11] Robust convex optimisation, on the other hand, would be a valid option to develop for visual odometry for which image-base measurements are associated with uncertainties. The common way of expressing this uncertainty information is in terms of covariance matrices.

Extracting feature points is a first step in many vision applications such as 3D reconstruction or camera resectioning. Uncertainty on detected feature points have well been investigated in the literature.[11–13] Many studies tried to improve the optimisation problems by incorporating these uncertainty information. Brooks *et al.* showed that covariance may be used to improve the quality of the estimate of the fundamental matrix. Similar results were concluded by Zeisl *et al.* However, Kanazawa and Kanatani[12] stated that covariance matrices for image features based on the calculate Hessian information do not improve homography or fundamental matrix estimation since they are isotropic and of similar size. On the other hand, Zeisl *et al.*[13] showed a performance improvement for bundle adjustment by incorporating the uncertainty for scale invariant feature points.

Haralick describes how to propagate additive random perturbations through the various stages of a vision based algorithm.[14] A similar idea is adopted by Leo *et al.* who present a methodology for the propagation of the measurement uncertainty from initial stages through the stereo calibration to the uncertainty in triangulation.[15] However, a complete method using covariance information to get better estimates of the camera motion for monocular systems has not been given.

2. The Proposed Solution

Following from this, we aim, in this work, to robustly improve the camera motion estimations of an innovative monocular visual odometry solution by incorporating feature localisation uncertainties and their propagation through the multiple view geometry algorithms. Our approach provides a framework to obtain robust and global solution under the L_∞ norm. This approach, which relies on modern optimisation methods, is more efficient in dealing with uncertain data than their traditional gradient-based counterparts; therefore a robust and global solution is guaranteed.

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

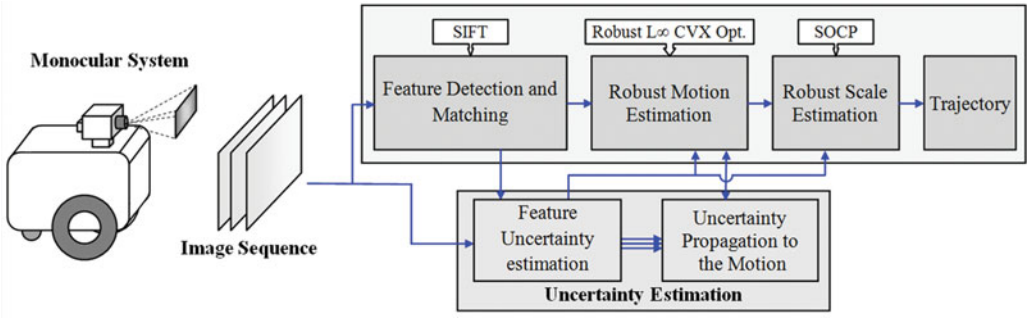


Fig. 1. A block diagram showing the main architecture of the proposed solution.

Although most researchers avoid uncertainty due to the added complexity in constructing the robust optimisation model and to the lack of knowledge of the nature of these uncertainties, especially their propagation, our work focuses on robust convex optimisation along with estimating the uncertainties in every step of the algorithm, starting from uncertainties in features positions. First, we propose a technique for minimising errors by incorporating these uncertainties from all sources and their propagation to the rotation, to the translation and to their corresponding 3D scene points estimates, using robust L_∞ convex optimisation via the Second-Order Cone Programming (SOCP). Secondly, we propose to use the robust least squares solution via the SOCP as well capable of dealing with system uncertainties for frame to frame absolute scale estimation. The proposed solution, which is geometrically meaningful, does not possess any element of randomisation in which the global optimality is ensured since their local minimum are, by definition, guaranteed to be a global minimum.

Thus, our first set of contributions consists of globally robust and optimal solutions to problems that arise in monocular motion estimation using robust convex optimisation under the L_∞ norm. Our goal is to provide a solution with a priori guaranteed feasibility when the uncertain problem's parameters vary within the approved uncertainty set defined after propagation. Implementation of these techniques is conducted on challenging real data collected from indoor environment, urban environment and data gathered at a Mars/Moon analogue site.

The proposed solution, described in Fig. 1, assumes a fully calibrated system with known intrinsic parameters K. Using a vehicle equipped with a single camera capturing sequences of images, the final goal is to estimate the camera pose at each time step relying only on these images and incorporating the uncertainties. Thus, the main steps of our algorithm are:

- Extraction of image feature points using SIFT detector and estimating their uncertainties;
- Estimating the initial relative rotations R_i and the translation T_i via the essential matrix;
- Estimation of the propagated uncertainties to R_i and T_i through the normalized 8-point algorithm and the SVD;
- Estimation of the 3D scene points using convex optimisation along with their uncertainties;
- Optimising the motion using robust L_∞ convex optimisation taking in consideration all sources of uncertainties with a sequence of camera resectioning /triangulation;
- Computing the unknown absolute scale ratio using robust least squares via SOCP;

The rest of the paper is structured as follows. Basic concepts of robust convex optimisation are discussed in Section 3. Section 4 provides details on uncertainty estimation in features position using both Harris corner detector and SIFT estimator. In Section 4, we describe the propagation of uncertainties to the rotation and to the translation. In Section 6 we present techniques used in estimating uncertainties in the reconstructed 3D points. Robust L_∞ motion estimation and robust scale estimation algorithms are detailed in Sections 7 and 8 respectively. Results are given in Section 9 followed by Section 10 giving the main conclusions of this work.

3. Robust Convex Optimisation

Optimisation is essential in engineering and control design where most applications assume complete knowledge of the problem data to optimise. However, most optimisation problems deal with uncertain

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

data. Two main sources for the uncertainty are cited: data which are not exactly known or cannot be exactly measured, and the impossibility of the implementation of the exact solution due to inherent inaccuracy of the devices.[1] This data uncertainty results in uncertain constraints and objective function.

Robust optimisation is a recent approach to optimisation under uncertain data, where the uncertainty model is not stochastic, but rather deterministic. Even though it is still considered as a new approach to optimisation problems under uncertainty; more and more real applications have already proved its efficiency. Robust optimisation is mainly designed to allow uncertainty-affected optimisation problems to provide guarantees about the performance of the solution.[1,2] In other words, in this optimisation, instead of recovering the solution in some probabilistic sense under stochastic uncertainty, the optimiser builds a solution that is optimal for any realisation of the uncertainty in a given set.[18] In cases where the optimality of a solution is affected by the uncertainty, the robust optimisation main goal will be then to seek a solution that performs relatively well for any value taken by the unknown coefficients. While a common approach is to optimise the worst-case objective, more studies are conducted toward other robustness methods.[2]

Robust optimisation, in general form, deals with two sets of entities, decision variables and uncertain variables. Here, the first aim of worst-case robust optimisation is to recover the optimal solution on the decision variables such that the worst-case is minimised and the constraints are robustly feasible, while the uncertainty is allowed to take arbitrary values in a defined uncertainty set.[19] The optimal solution is evaluated using the realisation of the uncertainty that is most unfavourable.[2] The general form of this robust optimisation is given by:

$$\begin{aligned} &\min_{x} && \max_{\omega} \ f(x, \omega) \\ &\text{subject to} && g(x, \omega) \leq 0 \, ; \forall \omega \in \mathcal{W} \end{aligned} \tag{1}$$

Where ω is the uncertain variables, $\mathcal{W}$ the uncertainty set and x is the decision variables.

In the main part of this work, we deal with convex optimisation problems for monocular motion estimation for which the data are uncertain and known to belong to a given uncertainty set $\mathcal{W}$. These problems can be efficiently recast and solved using Second-Order Cone Programming (SOCP), which is a standard technique in convex optimisation. A SOCP constraint, which is a conic quadratic constraint, is of the form:

$$\| A_i x + b_i \|_2 \leq c_i^\top x + d_i, \ \ A_i \in \mathbb{R}^{(m) \times n} \ b_i \in \mathbb{R}^m c_i \in \mathbb{R}^n \ d_i \in \mathbb{R} \tag{2}$$

Where x is the variable vector and A_i,b_i,c_i,d_i are the constraints parameters.[8,20] The robust counterpart is the problem of finding x such that:

$$\| A_i x + b_i \|_2 \leq c_i^\top x + d_i, \ \ \forall (A_i, b_i, c_i, d_i) \in \mathcal{W}. \tag{3}$$

Boni *et al.* showed that a convex quadratic constraint with ellipsoidal uncertainty error can be implemented as a system of conic quadratic constraint.[1] In addition, they showed that, a conic quadratic constraint with ellipsoidal uncertainty error can be reformulated as a set of nearly conic quadratic constraints.

According to the standard in,[21] the uncertainty indicates the upper and lower values that an uncertain variable may assume after all systematic biases have been corrected. Suppose we have a measurement function g fulfilling some mathematical constraints with n uncertain input quantities $q_1, \ldots, q_n$, $p = g(q_1, , q_n)$, the uncertainty of the output could be estimated trough a first order Taylor approximation:[22]

$$u_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{\partial g}{\partial q_i} \right) \left(\frac{\partial g}{\partial q_j} \right) u(q_i, q_j) \tag{4}$$

Where $u(q_i, q_j)$ is the covariance of q_i and q_j and when $i = j$ the $\sqrt{(u(q_i, q_i))} = q_i$ is the uncertainty of q_i. Equation (4) can be written in a more general form:

$$u_p^2 = \underline{\Lambda}_g = \underline{J}_p \underline{\Lambda}_p \underline{J}_p^\top \tag{5}$$

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

(a) Indoor environment (b) Urban environment (c) Mars/Moon analogue

Fig. 2. Harris image features with location covariances visualised via error ellipses.

Where $\underline{\Lambda}_p$ is the input covariance matrix, and $\underline{J}_p$ is the input Jacobian matrix, namely the matrix of partial derivatives.

4. Feature Location Uncertainty

Extracting feature points is a first step in many vision applications such as Homography, 3D reconstruction or motion estimation. Since the detected feature points, regardless the nature of the detector, have some uncertainty, the proposed solution in this work to robustly estimate motions involves robust convex optimisation based on those uncertainties which are expressed in terms of covariance matrices.

4.1. Uncertainty estimation for Harris corner detector

Harris corner detector is the most famous algorithm in computer vision. For detection, this algorithm relies on the second order derivative matrix, or known as the second moment matrix, constructed from intensity values.

Assume $f(x, y)$ represent an image feature extracted using Harris corner detector.[21] The true location of this feature is given by $(\tilde{x}, \tilde{y})$. Errors in its location are then given by $(\Delta x, \Delta y) = (x - \tilde{x}, y - \tilde{y})$. If Δx and Δy are regarded as random variables, the covariance matrix is given as:[12]

$$\Lambda_x = \begin{bmatrix} E\left[\Delta x^2\right] & E\left[\Delta x \Delta y\right] \\ E\left[\Delta x \Delta y\right] & E\left[\Delta y^2\right] \end{bmatrix} \tag{6}$$

Where $E[\Delta.]$ denotes expectation. Two main techniques were used in literature for determining Λ_x, a residual based approach and a derivative based approach. These techniques are detailed in ref. [12]. We employ the latter for its ease of use and implementation.

We define a matrix of first order partial derivatives squared for spatial coordinates x and y as follows:

$$H = \begin{bmatrix} \sum_{(x,y)\in\mathcal{N}_p} \omega_{xy} I_x^2 & \sum_{(x,y)\in\mathcal{N}_p} \omega_{xy} I_x I_y \\ \sum_{(x,y)\in\mathcal{N}_p} \omega_{xy} I_y I_x & \sum_{(x,y)\in\mathcal{N}_p} \omega_{xy} I_y^2 \end{bmatrix} \tag{7}$$

where I_x and I_y denote the partial derivatives and ω_{xy} is a weighting function, normally Gaussian. The matrix H in (7), which is the Hessian matrix, describes the curvature distribution around a point. Therefore, the greater the change in curvature the more accurately the corner can be located and vice versa (Fig. 2). Hence, considering the inverse of this expression to define the covariance would be an acceptable approach:[12]

$$\Lambda_x = inv(H) \tag{8}$$

4.2. Uncertainty estimation for Scale Invariant Feature Transform (SIFT)

New algorithms that are not only able to detect points in an image but also interest regions[23] are developed to tackle the problems with Harris corner detector such as invariance to scale and rotation. These interest regions represent in general areas which are brighter or darker than the surrounding.[13] Popular region detectors are SIFT[23] and SURF.[24] We focus on evaluating localisation uncertainty for each region found using SIFT (Fig. 3). The latter detector uses Laplacian operator for spatial feature

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

(a) Indoor environment (b) Urban environment (c) Mars/Moon analogue

Fig. 3. SIFT image features with location covariances visualised via error ellipses.

detection and scale. Using the derivative based approach, location uncertainty for scale invariant feature points in SIFT is calculated as the inverse of the Hessian:[13]

$$\Lambda_x = \left(\sum_{(x,y)\in\mathcal{N}_p} \omega(i,j) \begin{bmatrix} D_{xx}(i,j,\delta_p) & D_{xy}(i,j,\delta_p) \\ D_{yx}(i,j,\delta_p) & D_{yy}(i,j,\delta_p) \end{bmatrix} \right)^{-1} \tag{9}$$

Where D_{xx}, D_{xy} and D_{yy} are the second order derivatives at point p, δ_p is the scale. $\mathcal{N}_p$ is the image point neighbourhood.

4.3. Feature localisation uncertainty implementation

Implementation of these techniques is conducted on challenging datasets and shown in Figures 2 and 3. The first one collected in our laboratory using a Pioneer P3-DX platform with a fully-calibrated forward looking camera. The second is gathered from a vehicle travelling in an urban city environment with a pointing-forward calibrated camera is mounted on this vehicle.[25] The third one is a collection of data from a Mars/Moon analogue site at Devon Island, Nunavut.[26]

Figures 4 and 5 show clearly that feature points localisation uncertainties using Harris corner detector are relatively less than those estimated using SIFT for all environments datasets. The results are summarised in Table I as well. The average error for Harris in urban environment, for example, is in the order of 0.04 whereas it reaches 0.15 using SIFT. In the Moon/Mars analogue environment and due to its nature, these uncertainties increase remarkably (0.05 for Harris corner detector and 0.25 for SIFT) which directly affects subsequent motion estimations. For the indoor environment, the same pattern is recorded, with an average error of 0.03 pixels for Harris and about 0.12 using SIFT, however in overall lower errors are noticed here in comparison to the two other environments and this is due to nature of the environment.

Matching in Harris corner detector algorithm is performed using the cross correlation between local image patches. This means that only features that correlate most strongly with each other in both directions are accepted. Therefore, as an important drawback, the matching accuracy and robustness, in this algorithm, is completely depending on the actual transformation between views. On the other hand, SIFT uses the Euclidean distance between two feature points vectors as the similarity criteria of the two key points and uses the nearest neighbour algorithm to match each other which increase significantly its accuracy. Even the matching using Harris corner detector can be performed with low time consumption, its accuracy is compromised comparing to the high accuracy and robustness matching that is provided by SIFT.

By analysing the obtained results from Harris corner detector and SIFT for both datasets, the latter is confirmed to provide relatively more stable and conservative uncertainty estimations. The later quality is necessary to discard underestimated uncertainty as in Harris which could influence on the performance of motion estimations. This in addition to matching accuracy of SIFT that justifies its adoption SIFT in our monocular motion estimation algorithm.

5. Feature Uncertainty Propagation in the Rotation Matrix and the Translation Vector

Our robust optimisation motion estimation algorithm takes as well into account the uncertainties in the rotation and the translation which are estimated through the propagation of feature position uncertainties via the 8-point algorithm and Singular Value Decomposition (SVD) algorithm.

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

Table I. Average localisation errors of the extracted feature points using Harris and SIFT on two real datasets.

	Average localisation errors	
	Harris	SIFT
Indoor site	0.03	0.12
Urban site	0.04	0.15
Moon/Mars analogue site	0.05	0.25

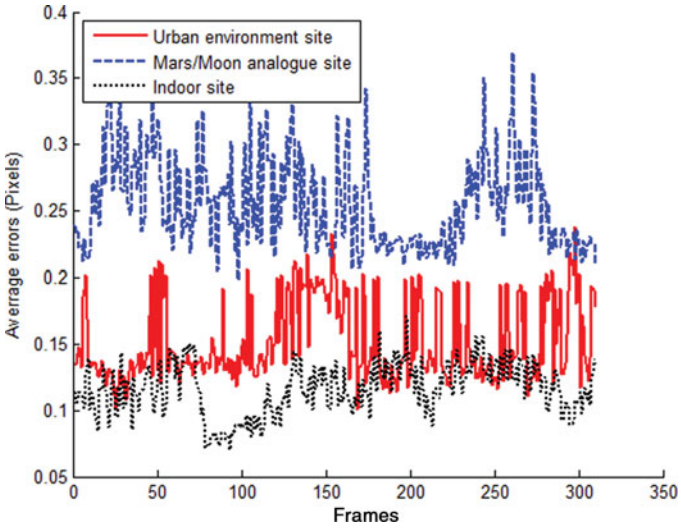


Fig. 4. Average feature points localisation errors using SIFT.

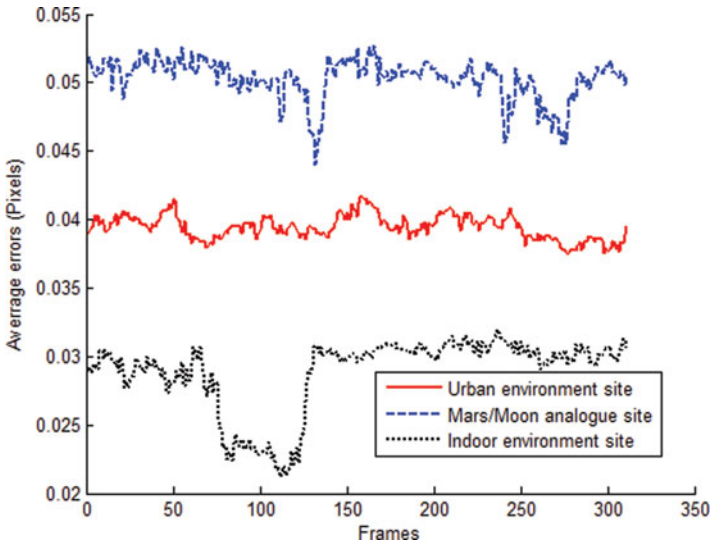


Fig. 5. Average feature points localisation errors using Harris corner detector.

It is clear that the entity encoding the translation and rotation comprising the 3D motion is the essential matrix E. E is defined by $E = [T]_\times R$, where T and R represent respectively the translation vector and the rotation matrix. In calibrated systems, this matrix is deduced from the fundamental matrix F which is estimated via the 8-point algorithm followed with an SVD process. Starting from the estimated uncertainties in feature points extraction, we present in this section techniques used in our implementation for estimating the uncertainty in the fundamental matrix, the uncertainty in the essential matrix and then the propagated uncertainties to T and R.

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

5.1. Uncertainty propagation in the fundamental matrix

The fundamental matrix is the crucial link that represents the vision geometry between two views in the pinhole camera model.[3] Given two views with camera matrices P_i and P_j, a pair of matching image points $x_i \Leftrightarrow x_j$ must satisfy: $x_j^\top F x_i = 0$ where F is the fundamental matrix. Estimating the fundamental matrix is the key stage for any motion estimation algorithm where information has to be retrieved from several images as a unique source. Since F is defined up to a scale factor, it can be estimated from no more than eight correspondences.[27] Since the subsequent motion estimation steps rely heavily on the estimation of this matrix, a rational interest on recovering its parameters should be allocated. Indeed, estimating an optimal F is a hard task since points locations are noisy and the correspondences are spoilt by outliers. RANSAC[28] is a well-known robust statistics solution for this type of problems.

Unfortunately, RANSAC and similar solutions are able to detect outliers, but the inaccuracy in the image point locations is still not estimated. In literature, two main methods are used for estimating the covariance of F: Monte-Carlo simulations, and the derivation of a closed-form formula. The uncertainty estimation method adopted in this work was originally introduced in ref. [27]. It is well known that for each pair of images I_i and I_j, point correspondences $\mathbf{x}_i \Leftrightarrow \mathbf{x}_j$ where $\mathbf{x}_i = (x_i, y_i, 1)$ and $\mathbf{x}_j = (x_j, y_j, 1)$, the 3×3 fundamental matrix F can be derived from the following system equation:

$$\mathbf{x}_j^\top F \mathbf{x}_i = (x_j, y_j, 1)^\top \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{11} & F_{32} & F_{33} \end{bmatrix} (x_i, y_i, 1) = 0 \tag{10}$$

Let the vector $\mathbf{f}$ made up of the entries F where: $\mathbf{f} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^\top$ If we have n correspondences, then the $n \times 9$ matrix $\mathbf{M}$ is given by:

$$\mathbf{M} = \begin{bmatrix} x_{j_1}x_{i_1} & x_{j_1}y_{i_1} & x_{j_1} & y_{j_1}x_{i_1} & y_{j_1}y_{i_1} & y_{j_1} & x_{i_1} & y_{i_1} & 1 \\ x_{j_2}x_{i_2} & x_{j_2}y_{i_2} & x_{j_2} & y_{j_2}x_{i_2} & y_{j_2}y_{i_2} & y_{j_2} & x_{i_2} & y_{i_2} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{j_n}x_{i_n} & x_{j_n}y_{i_n} & x_{j_n} & y_{j_n}x_{i_n} & y_{j_n}y_{i_n} & y_{j_n} & x_{i_n} & y_{i_n} & 1 \end{bmatrix} \tag{11}$$

The epipolar geometry constraint, that is,

$$\forall k \in [1, n], \mathbf{x}_j^\top F \mathbf{x}_i = 0$$

simply leads to the matrix equation $\mathbf{Mf} = 0$. The well-known method for solving problem is by using singular value decomposition (SVD) of $\mathbf{M}$ and putting the smallest singular value of F to zero (rank 2 constraint enforcement).

Using the propagation property of the uncertainty through the non-linear systems in (5), and as detailed in ref. [27], the basic outline of the algorithm for estimating the uncertainty of the fundamental matrix is described in the following steps:

- Computation of J_X, the Jacobian of $\tilde{\mathbf{f}}$, where $\tilde{\mathbf{f}} = \tilde{\mathbf{M}}^{-1}\mathbf{c}$ is the solution to the linear system of equations ($\tilde{\mathbf{M}}$ is the sub-matrix of M containing the first eight column and $\mathbf{c} = -[\mathbf{1}_8]^\top$ is an 8×1vector);
- Estimation of the covariance matrix $\Lambda_{\tilde{\mathbf{f}}} = J_x \Lambda_x J_x^\top$ where Λ_x is the covariance matrix representing the feature position uncertainties.
- Enforcing the rank 2 constraint by computing (SVD):

$$F = UD \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top$$

- Computation of J_{SVD}, the Jacobian of the SVD using techniques introduced in ref. [29];

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

- Finally, the 9×9 covariance of the fundamental matrix F is given by:

$$\Lambda_F = J_{SVD} \begin{bmatrix} \Lambda_{\tilde{\mathbf{f}}} & \mathbf{0}_{8,1} \\ \mathbf{0}_{1,8} & 0 \end{bmatrix} {J_{SVD}}^\top \tag{12}$$

5.1.1. Uncertainty propagation in the essential matrix. The relationship between the fundamental and the essential matrices is given by:[3]

$$E = K_j^\top F K_i \tag{13}$$

Where K is the camera calibration matrix. Using (5) and given the covariance of the fundamental matrix Λ_F, the covariance of E, Λ_E can be computed as:[29]

$$\Lambda_E = \frac{\partial \left(K_j^\top F K_i \right)}{\partial F} \Lambda_F \frac{\partial \left(K_j^\top F K_i \right)}{\partial F}^\top \tag{14}$$

The derivative of $K_j^\top F K_i$ with respect to the element F_{ij} of F is equal to:

$$\frac{\partial \left(K_j^\top F K_i \right)}{\partial F} = K_j^\top \frac{\partial F}{\partial F} K_i \tag{15}$$

5.2. Uncertainty propagation in R and T
It is clear that given the essential matrix

$$E = U diag(1, 1, 0) V$$

and making the first camera matrix $P_i = [I|0]$, then the four possible choices for the second camera matrix P_j are:[3]

- $P_j = [R_1| + T]$, $P_j = [R_1| - T]$,
- $P_j = [R_2| + T]$, $P_j = [R_2| - T]$.

Where the rotation matrices $R_1 = UWV^\top$ and $R_2 = UW^\top V^\top$, the translation vector $T = U(0, 0, 1)^\top$ is the last column of U and $W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

5.2.1. Estimation of the covariance of the rotation matrix. The uncertainty of the rotation matrix R_1 knowing Λ_E, the uncertainty of E is given by:

$$\Lambda_{R1} = \frac{\partial \left(UWV^\top \right)}{\partial E} \Lambda_E \frac{\partial \left(UWV^\top \right)}{\partial E}^\top \tag{16}$$

The same technique is used to estimate the covariance of R_2 where we use $W^\top$ instead of W.

5.2.2. Estimation of the covariance of the translation vector. The translation vector T is given by $u_3 = U(0, 0, 1)^\top$ the last column of U. Hence its covariance Λ_T is then simply:

$$\Lambda_T = \frac{\partial (u_3)}{\partial E} \Lambda_E \frac{\partial (u_3)}{\partial E}^\top \tag{17}$$

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

6. Feature Uncertainty Propagation in 3D Reconstruction

Assume we have m views of a scene point $\hat{\boldsymbol{X}}$ which maps to image points $x_i = [u_i, v_i]$ via camera matrices P_i, reconstruction is to recover the 3D space position of points $\hat{\boldsymbol{X}}$ such that $x_i = P_i\hat{\boldsymbol{X}}$ for $i = 1, \ldots, m$. These quantities are related by the projection function f where:

$$f_u(R, T, x_i, X) = u_i - \frac{{r_{i1}}^\top X + t_{i1}}{{r_{i3}}^\top X + t_{i3}} = 0$$
$$f_v(R, T, x_i, X) = v_i - \frac{{r_{i2}}^\top X + t_{i2}}{{r_{i3}}^\top X + t_{i3}} = 0 \tag{18}$$

Clearly the camera matrices $P_i = [R_i|t_i]$ where $R_i = [r_{i1}, r_{i2}, r_{i3}]$ and $t_i = [t_{i1}, t_{i2}, t_{i3}]$ are the rotation matrix and the translation vector respectively. Note that $\hat{\boldsymbol{X}} = [X, 1]$ is represented by homogeneous coordinates ($X = [\mathrm{X}, \mathrm{Y}, \mathrm{Z}] \in \mathbb{R}^3$). In calibrated system, normalized camera matrices are used, where $P_1 = [I|0]$, set as a reference camera, and $P_2 = [R_2|T_2]$. In this case, a system composed of four equations can be obtained from (18).

Let Λ_X represents the covariance matrix of the scene point X and $\Lambda_{In} = diag(\Lambda_{x_i}, \Lambda_{R_i}, \Lambda_{T_i})$ is the diagonal covariance matrix associated with the input parameters vector $In = [x_i, R, T]$ which was estimated using techniques presented in Sections 4 and 5. Then, the following expression can be written to model the covariance propagation:[30]

$$J_X \Lambda_X J_X^\top = J_{In} \Lambda_{In} J_{In}^\top \tag{19}$$

Where J_X and J_{In} are the Jacobian matrices of derivatives of f in (18) with respect to the 3D point X and the input parameters vector $In = [x_i, R, T]$ respectively:

$$J_X = \begin{bmatrix} \frac{\partial f_{u_1}(R,T,x_1,X)}{\partial X} & \frac{\partial f_{u_1}(R,T,x_1,X)}{\partial Y} & \frac{\partial f_{u_1}(R,T,x_1,X)}{\partial Z} \\ \frac{\partial f_{v_1}(R,T,x_1,X)}{\partial X} & \frac{\partial f_{v_1}(R,T,x_1,X)}{\partial Y} & \frac{\partial f_{v_1}(R,T,x_1,X)}{\partial Z} \\ \frac{\partial f_{u_2}(R,T,x_2,X)}{\partial X} & \frac{\partial f_{u_2}(R,T,x_2,X)}{\partial Y} & \frac{\partial f_{u_2}(R,T,x_2,X)}{\partial Z} \\ \frac{\partial f_{v_2}(R,T,x_2,X)}{\partial X} & \frac{\partial f_{v_2}(R,T,x_2,X)}{\partial Y} & \frac{\partial f_{v_2}(R,T,x_2,X)}{\partial Z} \end{bmatrix}$$

$$J_{In} = \begin{bmatrix} J_{x_1} & 0 & J_{RT_1} & 0 \\ 0 & J_{x_2} & 0 & J_{RT_2} \end{bmatrix}$$

$$J_{x_i} = \begin{bmatrix} \frac{\partial f_{u_1}(R,T,x_1,X)}{\partial u_i} & \frac{\partial f_{u_1}(R,T,x_1,X)}{\partial u_i} \\ \frac{\partial f_{v_1}(R,T,x_1,X)}{\partial u_i} & \frac{\partial f_{v_1}(R,T,x_1,X)}{\partial u_i} \end{bmatrix}$$

$$J_{RT_i} = \begin{bmatrix} \frac{\partial f_{u_i}}{\partial R_{i_{11}}} & \frac{\partial f_{u_i}}{\partial R_{i_{12}}} & \cdots & \frac{\partial f_{u_i}}{\partial R_{i_{33}}} & \frac{\partial f_{u_i}}{\partial T_{i_1}} & \cdots & \frac{\partial f_{u_i}}{\partial T_{i_3}} \\ \frac{\partial f_{v_i}}{\partial R_{i_{11}}} & \frac{\partial f_{v_i}}{\partial R_{i_{12}}} & \cdots & \frac{\partial f_{v_i}}{\partial R_{i_{33}}} & \frac{\partial f_{v_i}}{\partial T_{i_1}} & \cdots & \frac{\partial f_{v_i}}{\partial T_{i_3}} \end{bmatrix}$$

The output covariance matrix and, thus, the output uncertainties are given by:

$$\Lambda_X = {J_X}^\dagger (J_{In} \Lambda_{In} J_{In}^\top) {J_X}^{\top^\dagger} \tag{20}$$

Where ${J_X}^\dagger = (J_X^\top J_X)^{-1} J_X^\top$ is the pseudo inverse matrix of J_X. An extensive experimental validation has been conducted of the developed theory above. Figure 6 shows the 3D points' uncertainties patterns of a sequence frame against their respective depths. It can be clearly seen that closer points have smaller uncertainties than relatively distant points. Therefore, tracked points over long sequence of images will have decreasing uncertainties as the vehicle approaches them. Note that while these uncertainties are dependent to the keypoints' image positions uncertainties, they are completely independent of their images coordinates (u_i, v_i). Indeed, selecting relatively closer points would be better for good robust motion estimation.

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

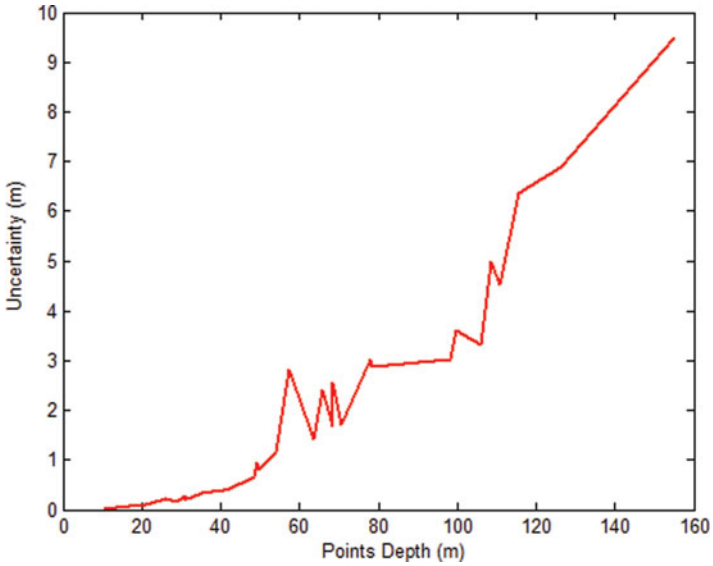


Fig. 6. 3D points uncertainties against their depths.

As a summary of Sections 4–6:

- Uncertainties in features positions are given in (8) and (9);
- Propagated uncertainties to the rotation and the translation are given in (16) and (17) respectively;
- Propagated uncertainty to the reconstructed 3D points is given in (20).

7. Robust L_∞ Motion Estimation Algorithm

After having estimated Λ_x, the uncertainties in features position from (8) and (9) and uncertainties in the initial rotation and translation Λ_R and Λ_T from (16) and (17) respectively, robust L_∞ motion estimation is then performed. For each consecutive image pair, the whole algorithm is summarized in the following operations:

- Estimating the 3D position for each point X_i by Robust L_∞ triangulation using covariance matrices Λ_{x_i}, Λ_T and Λ_R(Section 7.2);
- Estimating the covariance matrix Λ_{X_i} for each recovered point X_i using (20) (Section 6);
- Estimating the camera parameters by robust L_∞ resectioning using covariance matrices Λ_{x_i} and Λ_{X_i} (Section 7.3).

These operations are repeated until the L_∞ projection errors reach a satisfactory minimum.

7.1. Computational cost

It is known that the computational complexity of the Levenberg-Marquardt algorithm is: $\mathcal{O}((m+n)^3))$, where m is the number of cameras and n is the number of 3D scene points.[3] In our algorithm, at a given time step we are dealing with either the triangulation problem or camera parameters recovering problem for which they are solved using bisection algorithm for feasibility checks. The triangulation problem part has a computational complexity of $\mathcal{O}(m^{1.5})$ and a memory requirement of $\mathcal{O}(m)$ where m is the number of cameras in which the triangulating point is visible.[9,31] Similarly, the camera parameters recovering problem has a computational complexity of $\mathcal{O}(n^{1.5})$ and a memory requirement of $\mathcal{O}(n)$ where n is the number of scene points. Therefore, for one whole update of the proposed solution when dealing with n scene points and m cameras the computational complexity is given by: $\mathcal{O}(mn(\sqrt{m}+\sqrt{n}))$ and the memory requirement is $\mathcal{O}(\max(m,n))$.[9]

Number of iterations as well is crucial in this kind of problems, when a bisection search is performed. In our solution defining the initial diameter of the second order cones plays a very important role in determining the number of required iterations. Hence, the upper and the lower parameters of the bisection algorithm are chosen so the search area is reduced. A memorable search based on the previous iteration parameters is performed where these parameters are chosen in relation

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

with the previous iteration results. This technique reduces significantly the number of iterations and hence the time consumption. This makes this solution comparable to the classical L_2 Bundle Adjustment (BA) in term of time consumption and in the same time it recovers the global minimum of the cost functions which is a huge advantage over the classical L_2 BA.

7.2. Robust L_∞ triangulation with uncertain data

In order to provide an optimal solution to subsequent L_∞ robust motion estimation, a need for a robust and efficient optimal triangulation algorithm will be crucial. Our approach is based on robust convex optimisation with L_∞ norm taking in consideration all sources of uncertainties. Our data uncertainty are bounded, therefore, we are looking for optimal solutions which are feasible for any realisation of the data.

As described in Section 6, the function f in (18) relates all parameters of the triangulation between two cameras positions P_i. In this problem, the aim is to recover the value of X which minimises the maximum of this re-projection error across all images:

$$\varepsilon_i = d(x_i, P_i\hat{\mathbf{X}}) \tag{21}$$

Where d denotes image-space Euclidean distances between the measured and the projected points. Our aim is then to recover X which minimises the maximum of this re-projection error across all images. Given the camera matrices $P_i = [R_i|t_i]$ where $R_i = [r_{i1}, r_{i2}, r_{i3}]$, $t_i = [t_{i1}, t_{i2}, t_{i3}]$, $\hat{\mathbf{X}} = [X, 1]$ and their corresponding images $x_i = [u_i, v_i]$; and by considering uncertainties in feature position Δx_i, in rotation matrices ΔR_i and in translation vectors ΔT_i from covariance matrices Λ_{x_i}, Λ_{R_i} and Λ_{T_i} respectively; the L_2 norm of this re-projection error function is given by:

$$F_i(X) = \left\| \begin{matrix} \left(u_i + \Delta_{u_i}\right) - \frac{(r_{i1}^\top + \Delta r_{i1}^\top)X + (t_{i1} + \Delta t_{i1})}{(r_{i3}^\top + \Delta r_{i3}^\top)X + (t_{i3} + \Delta t_{i3})}, \\ \left(v_i + \Delta_{v_i}\right) - \frac{(r_{i2}^\top + \Delta r_{i2}^\top)X + (t_{i2} + \Delta t_{i2})}{(r_{i3}^\top + \Delta r_{i3}^\top)X + (t_{i3} + \Delta t_{i3})} \end{matrix} \right\|_2 \tag{22}$$

This problem is formulated within a quasi-convex optimisation framework. By using the L_∞ norm instead, the projection error will be given by:

$$G(X) = \max_i F_i(X) \tag{23}$$

For a scene point X to be visible as image points x_i it must lie in front of all cameras P_i. This implies the constraint $g(X, P_i) > 0$ for all i, where $g(X, P_i) = (r_{i3} + \Delta r_{i3})^\top X + (t_{i3} + \Delta t_{i3})$. Our optimisation problem is then given by:

$$\begin{aligned} \min_x \quad & G(X) \\ \text{subject to} \quad & g(X, P_i) > 0\,; \forall i = 1, \ldots, m \end{aligned} \tag{24}$$

The m error residuals in (21) give the error vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_m)^\top$. The estimated scene point is then the vector X that minimises the norm of this error vector. Given an upper bound γ, the inequality $\varepsilon_i \leq \gamma$ defines a set representing a second order cone represented by the first equation of optimisation problem (24). Note that each projection defines a conical surface where the bound γ is the radius of this cone and the camera centre is its apex. It is worthy as well to mention here that each image measurement adds one conical constraint in (24). This optimisation problem is solved using a sequence of robust SOCP feasibility problem capable of dealing with features' uncertainties and computing robust solution.[19] This leads toward using a bisection search to find the minimum value of γ for which the optimisation problem is feasible. The recovered 3D points using this robust L_∞ triangulation and their uncertainties are then used in the optimisation algorithm for recovering the camera parameters.

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

7.3. Robust L_∞ camera resectioning with uncertain data

Assume we have n scene points $\hat{\mathbf{X}}_i$ with their uncertainties Δ_{X_i} which are mapped to image points x_i with again their uncertainties Δ_{X_i} via the camera projection matrix P. This mapping is written as $x_i = P\hat{\mathbf{X}}_i$ for $i = 1, \ldots, n$. The problem of camera resectioning is to find P given a set of correspondences $x_i \Leftrightarrow \hat{\mathbf{X}}_i$ for $i = 1, \ldots, n$ which minimises the maximum of the re-projection error across all points:

$$\varepsilon_i = d(x_i, P\hat{\mathbf{X}}_i) \tag{25}$$

Where d denotes again image-space Euclidean distances between two points in the image, the measured and the projected points. By introducing uncertainties in feature position $\Delta x_i = (\Delta u_i, \Delta v_i)$ and those in their 3D correspondence points ΔX_i; the L_2 norm of this re-projection error function is given by:

$$F(X) = \left\| \begin{matrix} \left(u_i + \Delta_{u_i}\right) - \frac{p^{1\top}(X_i+\Delta X_i)}{p^{3\top}(X_i+\Delta X_i)}, \\ \left(v_i + \Delta_{v_i}\right) - \frac{p^{2\top}(X_i+\Delta X_i)}{p^{3\top}(X_i+\Delta X_i)} \end{matrix} \right\|_2 \tag{26}$$

Where p^j denotes the j^{th} row vector of P. Similarly to the triangulation problem, the L_∞ re-projection error is a quasi-convex function of the unknown camera parameters and the global minimum can be obtained by solving the following optimisation problem:

$$\begin{aligned} &\min_x && \max_i F_i(X) \\ &\text{subject to} && p^{3\top}(X_i + \Delta X_i) > 0\,; \forall i = 1, \ldots, n \end{aligned} \tag{27}$$

The recovered camera parameters and their uncertainties are then used again in the subsequent robust L_∞ triangulation algorithm and so on as detailed in Section 7 above.

8. Robust Scale Estimation

After having robustly estimated the motion of the camera using robust L_∞ convex optimisation under uncertain data, ambiguities in the translation scale still occur. Unlike in the stereo scheme, the monocular visual odometry estimates both the relative motion and the 3-D structure up to an unknown scale. This absolute scale cannot be estimated unless information about real world is provided. Assuming we have i 3D points $\hat{\mathbf{X}}_i$ which maps to image points $\hat{x}_i = [u_i, v_i, 1]^\top$ via the normalized camera matrix $P = [R|T]$ then:[3,6]

$$n\hat{x}_i = [R|ST]\hat{\mathbf{X}}_i \tag{28}$$

Where S is the unknown scale factor[6] and n is the depth factor that takes into account the projection plan ambiguity. This leads to the problem of finding a solution S to the over determined set of equations:

$$\begin{aligned} (t_z u_i - t_x)S &= (r^1 - r^3 u_i)X_i \\ AS &= b \end{aligned} \tag{29}$$

Where r^i denotes the i^{th} row vector of R and $T = [t_x, t_y, t_z]^\top$. Finding a solution S to this problem in the least squares sense (LS) means minimizing the residual $\|\Delta b\|$ subject to $AS = b + \Delta b$.[6] However, knowing that x_i, X_i, R and T are subject to deterministic perturbations, this solution is expected to exhibit very sensitive behaviour to these perturbations.[32] Therefore thinking about using more robust estimator would be a valid option to investigate. Thus, in our implementation, robust least squares (RLS) technique is used instead.[16] Robust least squares (RLS) solution computes the exact value of the optimal worst-case residuals using again a convex second-order cone programming (SOCP) of

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

the general form:

$$\begin{aligned} \min_x \quad & f^\top x \\ \text{subject to} \quad & \|A_i x + b_i\|_2 \leq c_i^\top x + d_i, \ i = 1, \ldots, m. \\ & g_i^\top x = h_i, \ i = 1, \ldots, p. \end{aligned} \tag{30}$$

Where vectors $x, f, c_i, g_i \in \mathbb{R}^n$, scalars $d_i, h_i \in \mathbb{R}$, matrix $A_i \in \mathbb{R}^{(n_i-1)\times n}$ and $b_i \in \mathbb{R}^{n_i-1}$. The norm $\|.\|_2$ is the standard Euclidean norm $\|u\|_2 = (u^\top u)^{1/2}$. The SOCP that formulate this problem is:[16]

$$\begin{aligned} \min_x \quad & \lambda \\ \text{subject to} \quad & \|AS - b\|_2 \leq \lambda - \tau \\ & \left\| \begin{bmatrix} S \\ 1 \end{bmatrix} \right\|_2 \leq \tau \end{aligned} \tag{31}$$

The unique solution to this problem is then given by:

$$S = \begin{cases} \left(\mu I + A^\top A\right)^{-1} A^\top b & \text{if } \mu = (\lambda - \tau)/\tau > 0 \\ A^\ddagger & \text{else,} \end{cases} \tag{32}$$

Where $A^\ddagger$ is the pseudo-inverse matrix of A. (λ, τ) are the unique optimal solutions for problem (31).

9. Experimental Results

In order to test the proposed solution, we have used data from three different environments. The first one consists on data collected in our laboratory using a Pioneer P3-DX platform with a fully-calibrated forward looking camera. Ground-truth is collected from an OptiTrack motion-capture system that provides absolute position information with millimetre accuracy at 100 Hz (Fig. 7).

More challenging environments are used as well in our experiments. The second one is a dataset collected from a vehicle travelling in an urban environment with a pointing-forward calibrated camera mounted on the roof of this vehicle.[25] The third one is a collection of data gathered at a Mars/Moon analogue site at Devon Island, Nunavut.[26]

As we can realise, the three environments are completely different so the technique would be tested in unbiased circumstances. The concurrent methods of the proposed solution are those which use iterative optimisation via L_2 norm, hence comparisons with bundle adjustment method based on Levenberg-Marquardt algorithm on exactly the same data are given.

Thus, in this section, we show first the navigation results illustrating the motion estimation using robust convex optimisation applied on a vehicle travelling through a variety of environments. Second, the robustness of the proposed solution is investigated where our method is tested under different error-level scenarios. A sequence of robust SOCP feasibility problem for the convexity task using SeDuMi toolbox[33] is employed along with Yalmip[19] toolbox for uncertainties modelling.

9.1. Motion estimation

Motion estimation using robust convex optimisation algorithm was integrated into our implementation for each environment. As explained previously, uncertainties in feature positions and their propagation in the rotation, in the translation and in the reconstructed 3D points have been taken in consideration in this implementation. We compare the output of our solution to the classical bundle adjustment system. We illustrate that robust convex optimisation pose estimates are more accurate and more consistent under a diversity of circumstances in realistic ground vehicle scenarios. Figure 8 shows the performance of both algorithms on each environment. Figure 8(a) shows errors of indoor trajectory where the robot has performed two loops in the main room in our laboratory. Figure 8(b) plots errors of a trajectory of more than 350 meters estimated through 200 key-frames from the urban environment,

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

Fig. 7. Setup used in our indoor experimental validations.

while Fig. 8(c) shows errors of a travelled distance of more than 650 meters recovered through 1000 key-frames taken from the Mars/Moon analogue site.

It can be seen, from the Euclidian distance errors, that robust convex optimisation approach is more accurate in all environments than classical bundle adjustment using Levenberg-Marquardt approach. Indeed, convex optimisation has shown its ability to ensure the global minima in recovering the motion parameters in comparison to iterative methods where a predefined termination criterion is set which favours convergence to local minima.

Notwithstanding of convex optimisation properties, this can be explained as well by the fact that incorporating uncertainties made the optimisation problem more robust leading to an efficient min-max optimisation in presence of high levels of noise.

In addition to motion estimation, using robust algorithm for scale estimation in our monocular system ensured more consistent trajectories. Deployment of robust least squares under second-order cone programming (SOCP) along with incorporating uncertainties in both feature positions and in their corresponding 3D scene points has demonstrated its ability to compute the absolute scale accurately and robustly. This can be seen as well in the recovered trajectories in Fig. 9. This figure plots the trajectory estimates aligned with their corresponding ground truth. Figure 9(a) plots the trajectory of the indoor experiment while Fig. 9(b) gives the trajectories estimates of a portion of the Moon/Mars analogue dataset. Clearly, more accuracy can be distinguished when robust convex optimisation is used in comparison with classical bundle adjustment with LM algorithm. This is in accordance with the theory as the estimates should be globally optimal.

Achieved results are good, reaching errors smaller than 3% and normally bounded by $5 - 12\%$ in terms of travelled error, defined as:

$$\text{Travelled error} = 100.\frac{\text{abs(error)}}{\text{Travelled distance}} \tag{33}$$

This also shows that our algorithm is very much suitable for estimating the motion of a vehicle travelling in different environments where high level of noises of unknown nature are likely to occur.

9.2. Motion estimation robustness

We want to highlight the robustness of the proposed solution and its behaviour and sensitivity to high levels of noise. To do that, four scenarios were adopted in function of noise (Table II). In fact, we take the extracted image points $x_i = (u_i, v_i)$ using SIFT algorithm and then perturb them with varying levels of Gaussian noise Δx_i where $\hat{x}_i = (\hat{u}_i, \hat{v}_i) = x_i + \Delta x_i$. These noisy image points $\hat{x}_i$ are then subsequently used to estimate the camera motion and scene points X_i using the proposed solution (which we called here Robust CVX). The same noisy image points $\hat{x}_i$ are used as well to estimate the camera motion using a L_∞ convex optimisation without taking in consideration the uncertainties (which we called Normal CVX). Therefore, four different scenarios can be distinguished in this test as illustrated in Table II.

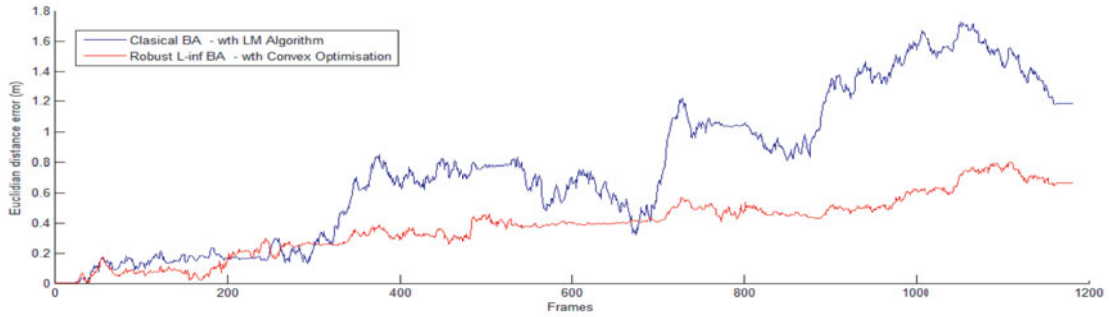
The first and second scenarios consist of using x_i and $\hat{x}_i$ respectively with the proposed robust L_∞ convex optimisation solution (Robust CVX) under uncertain data, while in the third and the fourth scenarios we use the same x_i and $\hat{x}_i$ respectively but with the L_∞ convex optimisation solution (Normal CVX) (Table II).

The estimated camera motions for the four scenarios are aligned with the ground truth and the Euclidian distance errors on camera position are computed. A plot of these errors is shown in Fig. 10.

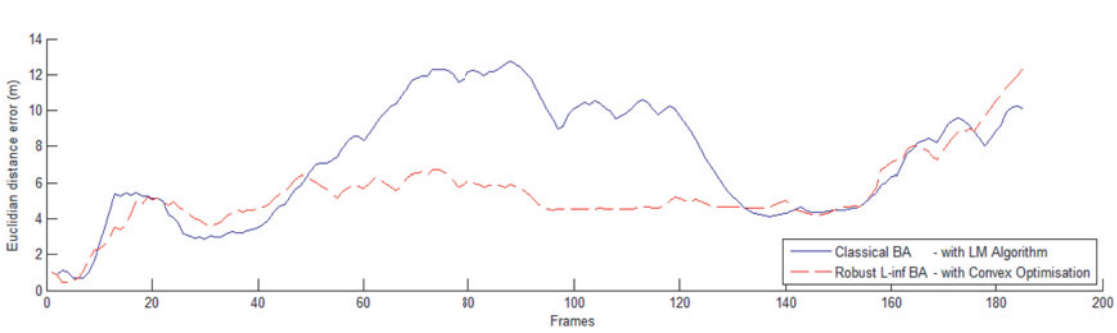
https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

Table II. The four scenarios for robustness investigation.

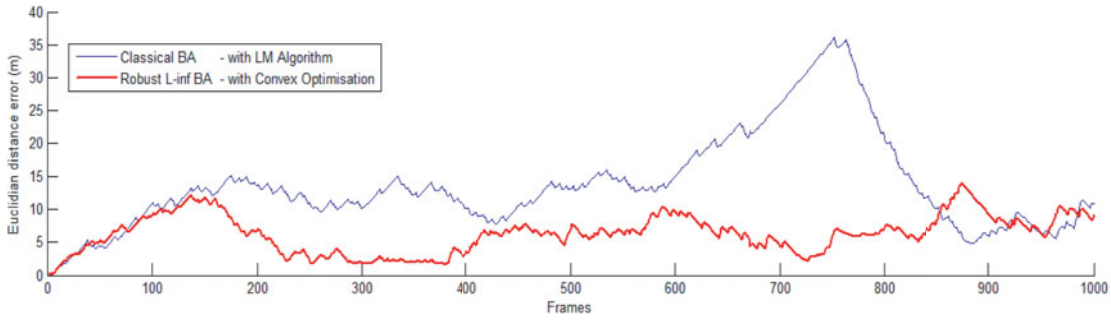
	Non-perturbed image points x_i	Perturbed,image points $\hat{x}_i$
Robust convex optimisation (Robust CVX)	Scenario 1	Scenario 2
Convex optimisation (Normal CVX)	Scenario 3	Scenario 4



(a) Indoor environment



(b) Urban environment



(c) Mars/Moon analogue

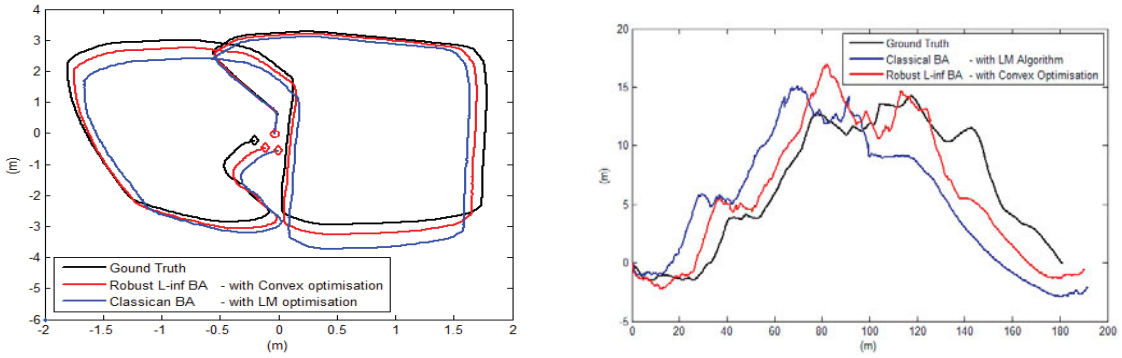
Fig. 8. Camera motion estimation errors.

This plot shows that motion can be estimated well even with a high level of noise using robust convex optimisation (Robust CVX) which is not the case when normal convex optimisation (Normal CVX) is used where important divergence is recorded (dashed blue line Fig. 10).

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

Table III. Average back projection errors of the estimation of the 3D scene points and the camera parameters of the two real datasets under the four scenarios.

	Average back projection errors		
	Indoor site	Urban site	Moon/Mars analogue site
Robust CVX	0.4985	0.7189	3.3107
Robust CVX + Gaussian noise	0.5102	0.7281	3.3204
Normal CVX	0.5215	0.7372	3.3401
Normal CVX + Gaussian noise	1.7253	2.0504	7.2978



(a) Indoor environment (b) Mars/Moon analogue

Fig. 9. Comparison between the trajectories estimates using robust convex optimisation and the classical BA using LM algorithm.

From these results, we learn that the proposed method, which takes in consideration the uncertainties of its inputs, performs remarkably better than the normal L_∞ convex optimisation as expected since it encodes large intervals in its optimisation and models well the uncertainties.

In order to assess the robustness in depth, back projection errors of the 3D scene points for the same scenarios are investigated as well. Figure 11 plots these projection errors of recovered 3D scene points on the image plan. These errors depict the accuracy of both the 3D position of the scene points and the camera motion parameters. Similar pattern to Fig. 10 can be noticed here as well. Typically robust convex optimisation performs well even when corrupted image points $\hat{x}_i$ are used and in all environments (Fig. 11 - solid red graphs). On the other hand, using normal convex optimisation for motion estimation with corrupted image points $\hat{x}_i$ would generate significant divergence (Fig. 11 - solid blue graphs). Logically both solutions provide similar performances with non-corrupted inputs x_i (dashed red and blue graphs). Indeed, it is clear that the uncertainty scheme always produces the best results.

Table III summarises the results regarding the average back projection errors for the different scenarios applied on the three environments. Obviously, the algorithm performs better in indoor environment. This can be justified by the fact that indoor features are more distinguishable (i.e. better matching) and their distances to the robot are relatively closer in comparison to features from the two other environments. Furthermore, the algorithm performs well in urban environment where feature points are relatively easily detected, matched and reconstructed in comparison to the Moon/Mars analogue site where the landscape presents more difficulties. Feature points extraction and matching is more challenging in this type of environments where their corresponding uncertainties are relatively higher as shown in Table I above. However the uncertainty incorporation impact is easily distinguishable in all environments, preventing potential divergence in presence of high level of noise.

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

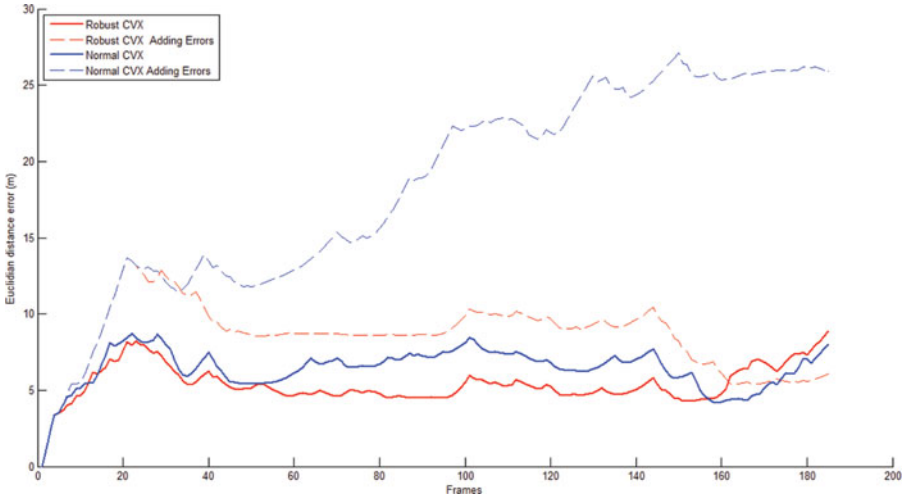


Fig. 10. Performance of the motion estimation algorithm as a function of noise level for the four different scenarios.

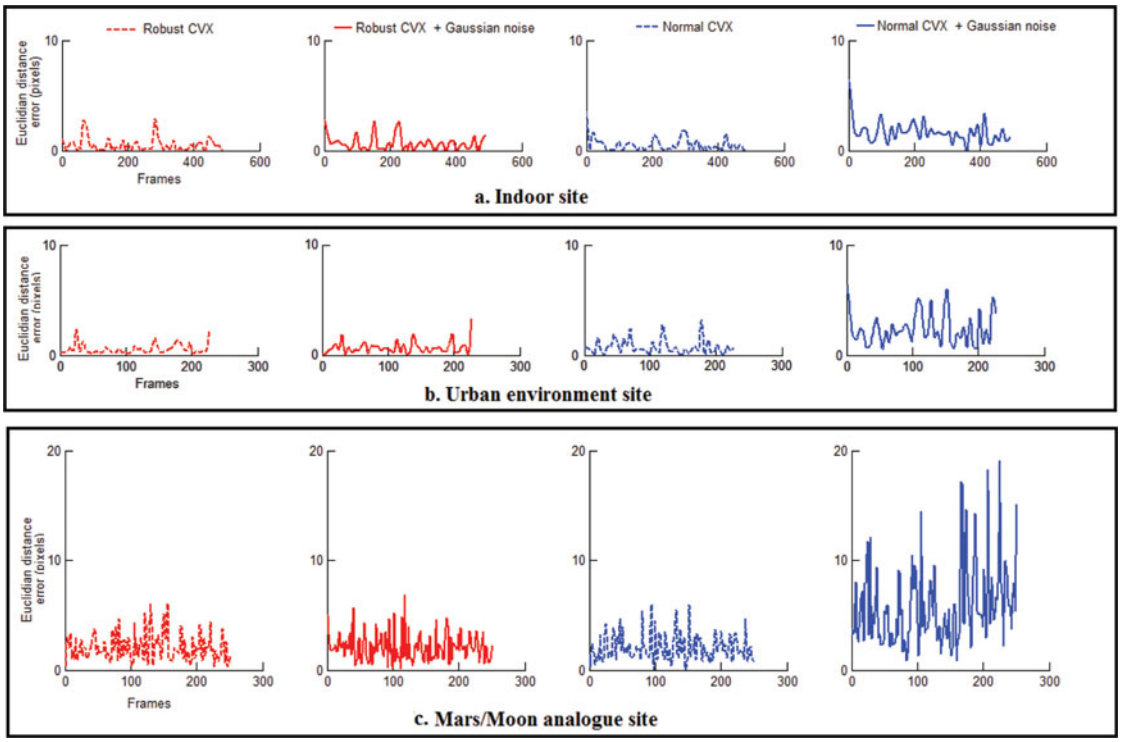


Fig. 11. Re-projection errors for the four different scenarios applied on the three environments.

10. Conclusions

In this work, a robust convex optimisation solution for monocular motion estimation systems has been presented. Including uncertainty estimation, based on SIFT derivative approach with the developed propagations through the eight-point algorithm and singular value decomposition SVD to the rotation and the translation of the camera and also to the 3D reconstructed points via triangulation, have proved improving global motion estimation. An experimental validation has been conducted and compared to optimal solution using classical bundle adjustment based on LM algorithm.

Although solutions to the motion estimation problem based on bundle adjustment with LM algorithm are eligible to provide accurate results, they present limitations in the presence of a high level of noise that a system based on robust L_∞ norm is able to overcome. Through several

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

experimental results, we show that our technique, by including all sources of uncertainties, clearly outperforms these classical techniques which use Levenberg-Marquardt algorithm for motion and Least Squares for absolute scale estimations.

Our second contribution, which follows on nicely from the first one, is to use the robust least squares algorithm capable of dealing with system uncertainties for frame to frame absolute scale estimation.

References

1. O. Boni, A. Ben-Tal and A. Nemirovski, "Robust solutions to conic quadratic problems and their applications," *Optim. Eng.* **9**, 1–18 (2007).
2. V. Gabrel, C. Murat and A. Thiele, "Recent advances in robust optimization: An overview," *Eur. J. Oper. Res.* **235**(3), 471–483 (2014)
3. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Edn. (Cambridge University Press, Cambridge, UK, 2004).
4. D. Nistr, O. Naroditsky and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics* **23**, 3–20 (2006).
5. D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.* **18**, 80–92 (2011).
6. I. Esteban, L. Dorst and J. Dijk, "Closed form solution for the scale ambiguity problem in monocular visual odometry," *Intell. Robot. Appl.* 665–679 (2010).
7. S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, UK, 2004).
8. F. Kahl, "Multiple View Geometry and the L$\imath$-norm," *Proceedings of the Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, 2 (2005) pp. 1002–1009.
9. K. Mitra and R. Chellappa, "A Scalable Projective Bundle Adjustment Algorithm using the L infinity Norm," *Proceedings of the Sixth Indian Conference Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08* (2008) pp. 79–86.
10. G. Chesi, "Estimation of the Camera Pose from Image Point Correspondences through the Essential Matrix and Convex Optimization," *Proceedings of the IEEE International Conference on Robotics and Automation, 2009. ICRA'09* (2009) pp. 35–40.
11. M. J. Brooks, W. Chojnacki, D. Gawley and A. Van Den Hengel, "What Value Covariance Information in Estimating Vision Parameters?." *Proceedings of the Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001*, Vol. 1 (2001) pp. 302–308.
12. Y. Kanazawa and K.-i. Kanatani, "Do we really have to Consider Covariance Matrices for Image Features?," *Proceedings of the Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001*, Vol. 2 (2001) pp. 301–306.
13. B. Zeisl, P. F. Georgel, F. Schweiger, E. G. Steinbach, N. Navab and G. E. R. Munich, "Estimation of Location Uncertainty for Scale Invariant Features Points." **In**: *BMVC* (2009) pp. 1–12.
14. R. M. Haralick, *Propagating Covariance in Computer Vision Performance in Characterization in Computer Vision* (Springer, Netherlands, 2000).
15. G. Di Leo, C. Liguori and A. Paolillo, "Covariance propagation for the uncertainty estimation in stereo vision," *IEEE Trans. Instrum. Meas.* **60**(5), 1664–1673 (2011).
16. L. El Ghaoui and H. Lebret, "Robust solutions to least-squares problems with uncertain data," *SIAM J. Matrix Anal. Appl.* **18**(4), 1035–1064 (1997).
17. D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Trans. Robot.* **24**(5), 1015–1026 (2008).
18. D. Bertsimas, D. B. Brown and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review* **53**(3), 464–501 (2011).
19. Johan Lfberg, "Automatic robust convex programming," *Optim. Methods Softw.* **27**(1), 115–129 (2012).
20. Qifa Ke and T. Kanade, "Quasiconvex optimization for robust geometric reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(10), 1834–1847 (2007).
21. Cordelia Schmid, R. Mohr and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.* **37**(2), 151–172 (2000).
22. G. Di Leo, C. Liguori and A. Paolillo, "Propagation of Uncertainty through Stereo Triangulation," *Proceedings of the Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE* (2010) pp. 12–17.
23. D. G. Lowe "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
24. H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up Robust Features," **In**: *Computer VisionECCV 2006* (Springer Berlin Heidelberg, 2006) pp. 404-417.
25. A. Geiger, M. Roser and R. Urtasun, "Efficient Large-scale Stereo Matching," **In**: *Computer VisionACCV 2010* (Springer Berlin Heidelberg, 2011) pp. 25–38.
26. P. Furgale, P. Carle, J. Enright and T. D. Barfoot, "The Devon Island rover navigation dataset," *Int. J. Robot. Res.* **31**(6), 707–713 (2012).

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press

27. F. Sur, N. Noury and M.-O. Berger, "Computing the Uncertainty of the 8 point Algorithm for Fundamental Matrix Estimation," **In**: *Proceedings of the 19th British Machine Vision Conference-BMVC 2008* (2008).
28. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**(6), 381–395 (1981).
29. T. Papadopoulo and M. I. A. Lourakis, "Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications," *Computer Vision-ECCV 2000* (Springer Berlin Heidelberg, 2000) pp. 554–570.
30. J. Chen, Z. Ding and F. Yuan "Theoretical uncertainty evaluation of stereo reconstruction," *Proceeding of the 2nd International Conference on Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008* (2008) pp. 2378–2381.
31. M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, "Applications of second-order cone programming," *Linear Algebr. Appl.* **284**(1), 193–228 (1998).
32. Lars Eldn, "Perturbation theory for the least squares problem with linear equality constraints," *SIAM J. Numer. Anal.* **17**(3), 338–350 (1980).
33. Jos F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.* **11**(1-4), 625–653 (1999).

https://doi.org/10.1017/S0263574714001829 Published online by Cambridge University Press