Covariance Estimation for Minimal Geometry Solvers via Scaled Unscented Transformation

Evren Imre*, Adrian Hilton

Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, GU2 7XH, UK

Abstract

Covariance is a well-established characterization of the output uncertainty for estimators dealing with noisy data. It is conventionally estimated via first-order forward propagation (FOP) of input covariance. However, since FOP employs a local linear approximation of the estimator, its reliability is compromised in the case of nonlinear transformations. An alternative method, scaled unscented transformation (SUT) is known to cope with such cases better. However, despite the nonlinear nature of many vision problems, its adoption remains limited. This paper investigates the application of SUT on common minimal geometry solvers, a class of algorithms at the core of many applications ranging from image stitching to film production and robot navigation. The contributions include an experimental comparison of SUT against FOP on synthetic and real data, and practical suggestions for adapting the original SUT to the geometry solvers. The experiments demonstrate the superiority of SUT to FOP as a covariance estimator, over a range of scene types and noise levels, on synthetic and real data.¹

Keywords: Covariance estimation, camera calibration, epipolar geometry.

---

¹The final publication is available at Elsevier via http://dx.doi.org/10.1016/j.cviu.2014.08.002.
1. Introduction

The output of a parameter estimator is of limited utility without a measure of its reliability. In practice, Gaussians are deemed an adequate uncertainty representation, and therefore, reliability is often assessed by propagating the input covariance to the output. However, since the output covariance is itself an estimate, its accuracy is a valid concern.

In this paper, we study the quality of covariance estimates computed via first-order propagation (FOP) and scaled unscented transformation (SUT), for a collection of common minimal geometry problems; namely, homography, epipolar geometry and calibration estimation, and triangulation. The associated solvers recover the mappings relating a set of image-image, or scene-image correspondences from a minimum number of correspondences, by using all available constraints. Their covariance is of interest in a number of practical applications. For example, in RANSAC [1], the covariance estimate computed from the minimal set with the best score is an upper bound to the actual uncertainty of the result. This is employed in [14] to approximate the measurement covariances in a prediction-correction scheme with an unscented Kalman filter (UKF), which estimates the internal and external calibration parameters of a moving camera. In [21], the covariance estimates are utilised within the RANSAC cycle to reject ambiguous/unstable solutions and to improve the hypothesis evaluation, resulting in significantly faster operation. It can also potentially be used for determining the search regions in guided matching [11], global registration of pairwise relative pose estimates [4], sensor fusion, and keyframe selection in structure-from-motion [26]. Finally, in film production, commercial software packages typically allow an operator to make use of known survey points for manually establishing correspondences for calibration [2]. A minimal geometry solver (MGS) requires less effort, and an estimate of the covariance can guide the operator to more reliable correspondences.

An MGS can be expressed as a function mapping a set of measurements (i.e. image-image, or scene-image correspondences) to a set of parameters (e.g. a camera pose). For such a map, covariance propagation via FOP involves substituting the original function with a local first order linear approximation [5]. Visual SLAM via extended Kalman filter (EKF) [8] is a well-known application of this approach, as FOP lies at the core of EKF. Other examples include the estimation of fundamental matrix [7] [22], 2D homography [6], and relative and absolute orientation [9].
The accuracy of FOP depends on the fidelity of the linear approximation to the original function. For example, EKF-SLAM is prone to underestimating the covariance (inconsistency) due to the accumulation of the linearisation errors [3]. An alternative approach is SUT [16]. It involves the generation of a number of “particles” by deterministically perturbing the original input, which are to be propagated through the function, in order to compute the sample covariance of the output. This technique approximates the output distribution by a Gaussian, as opposed to FOP, which approximates the function in such a way that the output distribution is a Gaussian [16]. SUT was recently employed in SLAM [12], by replacing EKF with UKF, and superior estimate consistency is reported. SUT is also utilised in [14] to estimate the uncertainty of a number of perspective-n-point (PnP) solvers.

The superiority of SUT over FOP is already established in the control and robotics literature (e.g. [16] [12]): it offers a performance identical to a second-order linear approximation (SOP). SUT is preferable to SOP as well, as it does not involve any Jacobians or Hessians [16]. However, the adoption of SUT in the computer vision field remains limited. In this paper, we aim to highlight the benefits of SUT on a collection of minimal geometry solvers. Our main contribution is twofold:

- We show that SUT is superior to FOP, by comparing the covariance estimates with the true covariance obtained through Monte-Carlo (MC) simulations in synthetic and real data.
- The original SUT is formulated for 1:1 functions mapping Euclidean vectors. We describe how SUT can be implemented when these conditions are violated, as is the case for most of the solvers.

The organization of the paper is as follows: Section 2 introduces the notation, and provides a brief background on FOP, MC and SUT. Section 3 discusses some solver-specific implementation details. The experimental results are presented in Section 4, followed by the conclusions in Section 5. Two appendices include further details on the solvers, and additional experiments.

2. Covariance Estimation for Nonlinear Transformations

Consider a solver $S$, an explicit function that maps a measurement vector $(\mathbf{x})$ to a vector representing the estimated parameters $(\mathbf{y})$. Given the measurement covariance matrix $(\mathbf{C}_x)$, our aim is to find an estimate $(\hat{\mathbf{C}}_y)$ to the output covariance $(\mathbf{C}_y)$. This can be accomplished either by linearising $S$ (FOP), or by computing the sample statistics of $\mathbf{y}$.
2.1. Covariance Estimation via Linearisation

The first two terms of the Taylor series expansion of $S(x)$ about an $\bar{x}$ provide a locally linear approximation to $S$ in $\Delta x = x - \bar{x}$, i.e.

$$ y = S(x) \approx S(\bar{x}) + J\Delta x, \quad (1) $$

where $J$ is the Jacobian of $S$ with respect to $x$, evaluated at $\bar{x}$. If $x$ is distributed as a Gaussian with the mean $\bar{x}$ and covariance $C_x$, i.e. $x \sim N(\bar{x}, C_x)$, the distribution of $y$ can be approximated as $N(S(\bar{x}), JC_xJ^T)$ [5]. This technique is known as first order (forward) propagation of covariance (FOP) [11].

$J$ can be computed analytically or numerically. The analytical method involves the algebraic derivation of a closed-form expression for $\nabla_x S$ (through the chain rule, when $S$ is a composition of several distinct stages). The numerical method computes one partial derivative at a time, by applying deterministic perturbations to each element of $\bar{x}$, and passing the results through a discrete high-pass filter. A common implementation, the central difference scheme, approximates the partial derivatives as

$$ \frac{\partial S}{\partial x_j} = \frac{S(x + \delta_j e_j) - S(x - \delta_j e_j)}{2\delta_j}, \quad (2) $$

where $\delta_j$ denotes the magnitude of the perturbation. $x_j$ is the $j$th element of $x$, and $e_j$ is a vector whose elements are all zeros, except for the $j$th, which is 1. The appropriate perturbation is problem-dependent; however, for geometry problems, [11] suggests $\max(10^{-6}, |10^{-4} x_j|)$ (pp. 602).

The analytical method has advantages in terms of accuracy and computational cost (a single evaluation of $\nabla_x S$, as opposed to $2M$ evaluations of $S$, for an $M$-element $x$). However, we adopted the numerical approach for the following reasons:

- We are not aware of any closed-form expressions of the Jacobians for some components of the solvers (e.g. Gröbner basis solvers).
- For a very common component, SVD, [32] offers a method to compute the exact Jacobian, but states that the results are identical to those obtained by the numerical approach.
- In practice, for geometry estimation problems, the error introduced by the numerical approximation is insignificant [11], especially, compared to the error due to the first-order approximation of $S$ (i.e. equation 1).
Therefore, we believe that this decision is unlikely to have a qualitative impact on the results.

A related method is **backward propagation of covariance** (BP): consider a function \( \hat{x} = T(y) \) that minimises \( ||x - \hat{x}||_{C_x} \), i.e. the Mahalanobis distance. If \( x \sim N(T(y), C_x)|_{y=\bar{y}} \), and \( J_T = \{ \frac{\partial T}{\partial y_j} \}_{j=1}^n |_{y=\bar{y}} \), \( C_y \) can be approximated as \( (J_T^T C_x^{-1} J_T)^{-1} \) [11]. When \( T \) is an analytical function, \( J_T \) can be computed without resorting to multiple evaluations. However, the applicability of this technique is limited to the cases where \( T(y) \) uniquely defines a vector in the measurement space, e.g. a \( T \) projecting a 3D scene point to the image plane of a camera (BP in structure-from-motion). Of the minimal solvers studied in this paper (Section 3), only the two-view triangulation case satisfies this condition, with \( T \) being the perspective projection operator, which maps a 3D point to two unique 2D image points, through two known camera matrices: alone, a camera matrix or an epipolar geometry relation is not sufficient to identify a unique scene-image or image-image correspondence, respectively. Furthermore, most solvers either do not minimise the Mahalanobis distance (e.g. epipolar geometry estimation), or perform a constrained minimisation (e.g. metric calibration estimation). Therefore, we will not consider BP further as an alternative linearisation method.

### 2.2. Covariance Estimation via Sample Statistics

Given a set of measurements \( \{x_k\} \sim N(\bar{x}, C_x) \), \( \hat{C}_y \) can be computed as the sample covariance of \( \{y_k\} = S(\{x_k\}) \),

\[
\hat{y} = \frac{1}{L} \sum_{k=1}^{L} u_k S(x_k), \\
\hat{C}_y = \frac{1}{L} \sum_{k=1}^{L} w_k (S(x_k) - \hat{y})(S(x_k) - \hat{y})^T
\]

where \( u_k \) and \( w_k \) are two sets of scalar weights, and \( L \) is the number of samples [5]. Both MC and SUT make use of equation 3, but differ in the way \( \{x_k\} \) is generated.

**Monte-Carlo Simulations:** In MC, \( \{x_k\} \) is randomly drawn from \( N(\bar{x}, C_x) \). \( u_k \) and \( w_k \) are uniformly set to \( \frac{1}{L} \) and \( \frac{1}{L-1} \), respectively. However, the simplicity of MC comes with two problems: redundant parametrisations and the computational complexity.
A parametrisation is redundant, when a $D$-dimensional entity is represented by $Q$ parameters, with $D < Q$ (e.g. a rotation matrix encodes the space of 3 dimensional rotations with 9 parameters: $D = 3$ and $Q = 9$). Since there are $D$ degrees-of-freedom, the rank of $C_y$ is at most $D$. However, MC does not enforce this constraint, and, the rank could be as high as $Q$. This issue can be remedied by converting the estimated parameters to a $D$-dimensional, minimal representation, as discussed in Section 3.

The computational complexity of MC is proportional to $L$, the number of samples in $\{x_k\}$. However, accurate estimates of $\bar{y}$ and $C_y$ often require a large $L$ [11]. Therefore, in terms of computational complexity, MC is usually impractical as a covariance estimator. However, since $\lim_{L \to \infty} \hat{C}_y(L) = C_y$, for sufficiently large $L$, it is considered a good approximation to the true statistics [5] [7] [11].

**Scaled Unscented Transformation:** SUT deterministically generates $\{x_k\}$ as a $L = 2M + 1$ element sample set, for an $M \times 1$ measurement vector [16]. Specifically, for the Cholesky decomposition

$$C_x = \Psi \Psi^T,$$  

the corresponding $\{x_k\}$ is constructed as [23]

$$x_1 = \bar{x}$$  

$$x_{k+1} = \bar{x} + \sqrt{\alpha^2(M+\kappa)}\Psi^k \quad k = 1...M$$  

$$x_{k+1+M} = \bar{x} + \sqrt{\alpha^2(M+\kappa)}\Psi^k \quad k = 1...M,$$

where $\Psi^k$ denotes the $kth$ row of the Cholesky factor $\Psi$. $\alpha$ and $\kappa$ are two scaling parameters, controlling the spread of $\{x_k\}$ around $\bar{x}$.

The weights $u_k$ and $w_k$ are calculated as follows [23]:

$$u_1 = 1 - \frac{M}{\alpha^2(M+\kappa)}$$  

$$w_1 = u_1 + (1 - \alpha^2 + \beta)$$  

$$u_k = w_k = \frac{1}{2\alpha^2(M+\kappa)}, k = 2...(2M+1),$$

where $\beta$ is a parameter used to incorporate prior information about known higher order statistics.

[16] and [23] discusses the selection of $\alpha, \kappa$ and $\beta$, and suggest optimal values for $x \sim N(\bar{x}, C_x)$ (i.e. the case of interest for this work):
• [23] states that $\kappa \geq 0$ guarantees a positive semi-definite $\hat{C}_y$, however the specific value is not critical. Therefore, it recommends $\kappa = 0$.

• $\alpha$ should be in $[0, 1]$. [23] recommends a small value when the transformation (i.e. $S$) is highly nonlinear.

• It is important that $\{x_k\}$ accurately characterises the distribution of $x$. When $x \sim N(\bar{x}, C_x)$, the moments of $\{x_k\}$ and $x$ match up to the 3rd order [16]. Setting $u_1 = 1 - \frac{M}{3}$ ensures a match in some of the 4th order moments [16]. It is also shown in [16] that $\beta = 2$ minimises the error in the 4th order moments.

• If $\kappa = 0$, $u_1 = 1 - \frac{M}{3}$ implies that $\alpha = \sqrt{\frac{3}{M}}$ (equation 6).

Therefore, throughout the paper, we follow the recommendations for the Gaussian case, and set $\kappa = 0$, $\alpha = \sqrt{\frac{3}{M}}$ and $\beta = 2$. We specifically indicate the cases where a smaller $\alpha$ is chosen due to the nonlinearity of the solver.

The computational cost of SUT is $2M + 1$ evaluations of $S$ (as opposed to $2M$ of FOP), plus the Cholesky decomposition of an $M \times M$ matrix. However, under the assumption of independent measurements, $C_x$ is either diagonal, or block-diagonal with $2 \times 2$ and $3 \times 3$ submatrices; therefore, the cost of the Cholesky decomposition is negligible.

3. SUT for Minimal Geometry Solvers

The formulation in Section 2 is not suitable for all minimal geometry solvers covered in this paper. Below is a presentation of the solver-specific issues, and implementation guidelines. Since all cases involve either image-image (2D-2D), or image-scene (2D-3D) correspondences as input, $\bar{x}$ is formed by concatenating the individual coordinates. The dimensionalities of the measurement and the parameter space are presented in Table 1.

In order to maintain the focus on covariance estimation, we choose to omit a detailed discussion of the solvers. However, each section provides the relevant references for the interested reader, and a brief exposition is included in Appendix A. Unless otherwise stated, the parameters for SUT are set to the values mentioned in Section 2.2, and those for the minimal solvers, to the defaults recommended by the original authors.

3.1. 2-View Triangulation

2-point triangulation (T2) solver estimates the 3D coordinates of a scene point, given its 2D projections on the image planes of two known cameras.
Table 1: The number of correspondences and, the measurement and parameter space dimensionalities of the minimal geometry solvers.

<table>
<thead>
<tr>
<th>Solver</th>
<th>T2</th>
<th>H4</th>
<th>P6P</th>
<th>F7</th>
<th>F8</th>
<th>E5</th>
<th>P2P</th>
<th>P2Pf</th>
<th>P3P</th>
<th>P4P</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Corr.</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>dim(x)</td>
<td>4</td>
<td>16</td>
<td>30</td>
<td>28</td>
<td>32</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>dim(y)</td>
<td>3</td>
<td>8</td>
<td>11</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

The solver we employ is the optimal triangulation algorithm [11]. \( y \) is the triangulated scene point. The application of SUT and FOP is straightforward: it only requires the computation of the 3D points corresponding to the various perturbations of the original pair of image points (as discussed in Section 2).

3.2. Epipolar Geometry Estimation

An epipolar relation \( Y \) defines a constraint of type \( p^T Y r = 0 \), and describes the epipolar geometry between the images to which the image correspondence \((p; r)\) belongs [11]. When the internal calibration parameters are unknown, \( Y \) is the fundamental matrix, and can be estimated via the 8-point (F8)\(^2\), or the 7-point (F7) method, which involves 8 and 7 image correspondences, respectively [11]. Otherwise, one solves for the essential matrix from 5 image correspondences (E5), via [19].

\( Y \), being a 3x3 matrix, is a redundant representation of the underlying entity: a fundamental matrix has 7 degrees of freedom, whereas an essential matrix has only 5. In order to avoid a singular covariance matrix, \( Y \) must be reparametrised by a minimal representation.

A 7-element minimal parametrisation for fundamental matrix can be obtained through the singular value decomposition (SVD) of \( Y \), as proposed in [25]: given the decomposition \( Y = USV^T \), \( U \) and \( V \) are rotation matrices, each of which can be encoded by a 3-parameter axis-angle vector. Since a non-degenerate fundamental matrix must be of rank-2, \( S \) is a diagonal matrix with two non-zero entries. However, since \( Y \) is a homogeneous entity, \( i.e. \ Y \equiv cY \ \forall c \in \mathbb{R} \setminus \{0\}, \) it can be scaled so that the largest singular value is 1. Therefore, a fundamental matrix can be represented by 2 axis-angle vectors and a scalar. Computation of the sample statistics of parametrisations with a rotation component is discussed in Section 3.4.

\(^2\)The 8-point algorithm is not a true minimal solver, as it does not use all available constraints.
A non-degenerate essential matrix can be decomposed into a rotation matrix, and a rank-2 skew-symmetric matrix representing a unit-norm translation vector [11]. This decomposition forms the basis of a minimal representation: similar to the fundamental matrix case, the rotation component can be represented by 3 parameters, as an axis-angle vector. The translation component is a vector on the 3D unit sphere, and therefore, can be encoded by its azimuth and elevation angles. When the elevation is $\pm \frac{\pi}{2}$, the azimuth can take any arbitrary value, which is chosen as 0.

The epipolar geometry case poses an additional challenge to SUT and FOP: both methods assume that $S$ is a one-to-one transformation. However, the solutions to the F7 and E5 algorithms are the roots of a 3rd and a 10th degree polynomial, respectively: there could be as many solutions as the real roots. We identified two strategies for such one-to-many solvers:

- All solutions satisfy the epipolar constraint for $\bar{x}$. However, barring degeneracies, only one can represent the epipolar geometry between the images. Therefore, a small set of additional matches can be used as a validation set, to identify the correct solution. This approach, effectively, makes a one-to-many solver one-to-one.
- Alternatively, one can estimate a separate covariance for each solution. However, it may not be obvious with which covariance a solution should be associated. Moreover, the number of real solutions may vary for different perturbations of $\bar{x}$.

In practice, often the epipolar geometry is estimated from a redundant set of correspondences; therefore, in our experiments, we chose the former approach.

### 3.3. Homography Estimation

This class of problems aims to find a mapping $Y$ that satisfies the projective equality relation $p \approx Yr$ for a homogeneous point correspondence $(p, r)$. $Y$ is estimated via the normalised direct linear transform algorithm (DLT) [11]. The instances we study are resectioning (i.e. projective camera matrix estimation) from 6 scene-image correspondences (P6P), and 2D homography from 4 image correspondences (H4).

Since $Y$ is a homogeneous transformation, identifying $y$ with the elements of $Y$ leads to a redundant parametrisation, and a singular covariance matrix. [27] proposes to mitigate this problem by characterising the uncertainty as
Figure 1: Rank histogram of the MC covariance matrices for two minimal parametrisations. Unit sphere is the case where $\mathbf{Y}$ is projected on a $D$-dimensional unit sphere. Decomposition denotes the decomposition of $\mathbf{Y}$ into SVD components or metric camera parameters, for H4 and P6P, respectively. Left: H4. Right: P6P

A Lie-normal distribution. Although this distribution has a functional form that is superficially similar to a Gaussian, we deem it outside of the scope of this paper, which is limited to the covariance estimation, i.e. Gaussian characterisations. A more relevant, and common non-redundant representation involves the removal of the redundancy via a projection to a plane tangent to the unit sphere at the vector representation of $\mathbf{Y}$ [11][24]. Despite its success in 2D- and 3D-point cases [24], in our experiments, due to numerical issues, this approach occasionally resulted in rank-deficient covariance matrices: Fig. 1 depicts the rank histogram for the H4 and P6P problems for a set of covariance matrices computed via the procedure described in Section 4.1.1 (steps 1-3), by using the parameter set $N1$ (Table 3) and the MC technique. As shown in Table 1, the rank of valid covariance matrices for H4 and P6P are 8 and 11, respectively.

An alternative representation for H4 is inspired by the fundamental matrix case, discussed in Section 3.2: a homography can be decomposed into two rotations and two scalars, obtained via the SVD of the homography matrix. Unlike a fundamental matrix, there is no rank constraint for a homography matrix. Therefore, the scalars are the second and the third largest singular values, which are normalised with the largest singular value.

As for the P6P, a projective camera matrix can be decomposed into a
Table 2: Calibration estimators. “K” indicates the necessary calibration parameters to run the solver, whereas “U” indicates what the solver returns. “DOF” indicates the number of unknown parameters per calibration component.

<table>
<thead>
<tr>
<th>Component</th>
<th>DOF</th>
<th>P2P</th>
<th>P2Pf</th>
<th>P3P</th>
<th>P4P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientation</td>
<td>3</td>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Position</td>
<td>3</td>
<td>K</td>
<td>K</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>Focal length</td>
<td>1</td>
<td>K</td>
<td>U</td>
<td>K</td>
<td>U</td>
</tr>
<tr>
<td>Lens distortion</td>
<td>1</td>
<td>K</td>
<td>K</td>
<td>K</td>
<td>U</td>
</tr>
</tbody>
</table>

set of intrinsic and extrinsic parameters [11]. Although this may yield an improbable camera, with skew and aspect ratio exhibiting significant deviations from 0 and 1, respectively, the decomposition is an algebraically valid minimal representation.

Both alternative parametrisations are equivalent to their “tangent-plane” counterparts, in that they represent the same entity. However, due to the numerical issues, they differ in their suitability for the covariance estimation problem: Fig. 1 indicates that, unlike the “tangent-plane” parametrisation [24], the decomposition-based parametrisations effectively mitigate the rank-deficiency problem. It should be noted that, since both methods operate on the same data, the observed difference cannot be explained by a degeneracy in the homographies involved.

Upon decomposition, FOP and SUT can be employed without any issues, to obtain a full-rank $\hat{C}_y$.

3.4. Calibration Estimation

A calibration estimator utilises the projective equality relation $p \approx Y(K, r)$ between the scene-image correspondence pair $(r; p)$, induced by the projection operator $Y$, to recover $K$, the unknown internal and/or external calibration parameters. What $K$ represents defines a sequence of progressively more difficult problems: 2-point orientation (P2P) [14]; 2-point orientation and focal length (P2Pf) [14]; 3-point orientation and position (P3P) [10] [13]; and 4-point orientation, position, focal length and lens distortion coefficient (P4P) [15] (Table 2). All solvers except for P2P involve polynomial equations of varying degrees, hence yield multiple real solutions (up to 2, 4 and 12, respectively). There are two more issues specific to the calibration estimation problem: sample statistics of the orientation component, and the selection of $\alpha$ (equation 6).

Consider the P2P case, where the only output is the orientation. The use of equation 3 is not appropriate, as neither $\hat{y}$, nor $S(x_k) - \hat{y}$ is guaranteed
to represent a valid rotation. Instead, $\hat{y}$ can be calculated as [18]

$$q_k = S(x_k)$$

$$\hat{y} = \arg \min_q \sum_k u_k \|R(q_k) - R(q)\|_F^2,$$  

(7)

where $q$ and $q_k$ are quaternions, $R(q)$ is the rotation matrix corresponding to $q$, and $\|\cdot\|_F$ is the Frobenius norm operator. As for the covariance, the difference quaternion is computed by inverse-rotation, and then converted to the axis-angle representation, to obtain a non-singular covariance [17]:

$$a_k = A(q_k \otimes \text{conj}(\hat{y}))$$

$$\hat{C}_y = \sum_k w_k a_k a_k^T,$$  

(8)

In (8), $\otimes$ denotes the quaternion multiplication operator and $\text{conj}(.)$, the quaternion conjugate. $A$ converts a quaternion to an axis-angle vector.

In other solvers, in which the orientation is only a part of the output, the mean orientation and $\{a_k\}$, the set of orientation differences, are computed separately, and concatenated to the other components (e.g. focal length in P2P, and position vector in P3P) for use in equation 3. Since $u_k$ must be non-negative [18], $\alpha = 1$.

Another issue with $\alpha$ stems from the smoothness of the manifold on which $\hat{x}$ reside. $\alpha$ determines the magnitude of the perturbations. If the manifold is smooth, the perturbed points remain on the manifold. However, in some cases, the perturbation may result in an $x$, for which no real solution exists. In our experience, only P4P is susceptible to this problem, which can be mitigated by picking smaller $\alpha$.

4. Experimental Results

4.1. Synthetic Data Experiments

4.1.1. Experimental Setup

The performance of SUT and FOP are compared against the true covariance, estimated via MC, by generating random problem instances on synthetic data. The following procedure is repeated 1000 times for P4P, and 10000 times for the other solvers (due to the high computational cost of P4P):
Table 3: Parameters for random problem generation. \( U(a,b) \) represents a uniform distribution between \( a \) and \( b \).

<table>
<thead>
<tr>
<th>Scene (m)</th>
<th>Camera (m)</th>
<th>Scene Noise Standard Deviation (m)</th>
<th>Image Noise Standard Deviation (pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor</td>
<td>20x4x20</td>
<td>4x2x4</td>
<td>0.01</td>
</tr>
<tr>
<td>OWB</td>
<td>400x30x400</td>
<td>20x3x20</td>
<td>0.1</td>
</tr>
<tr>
<td>ONB</td>
<td>400x30x400</td>
<td>4x2x4</td>
<td>0.1</td>
</tr>
<tr>
<td>N1</td>
<td>20x4x20</td>
<td>4x2x4</td>
<td>0.01</td>
</tr>
<tr>
<td>N4</td>
<td>20x4x20</td>
<td>4x2x4</td>
<td>0.02</td>
</tr>
<tr>
<td>N9</td>
<td>20x4x20</td>
<td>4x2x4</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Orientation (rad) \( \sim U(0, 2\pi) \)
Focal length (pixel) \( \sim U(1500, 5000) \)
Distortion coefficient (division model [15]) \( \sim U(-10^{-6}, 10^{-6}) \).

This is equivalent to a distortion ranging from -450 to 500 pixels on the image.

Table 4: Number of cameras and scene points generated for each problem instance.

<table>
<thead>
<tr>
<th>Solver</th>
<th>T2</th>
<th>H4</th>
<th>P6P</th>
<th>F7</th>
<th>F8</th>
<th>E5</th>
<th>P2P</th>
<th>P2Pf</th>
<th>P3P</th>
<th>P4P</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Cameras</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>#Points</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

1. Randomly generate 1-2 cameras, and a number of scene points, as dictated by the problem type (Table 4). For H4, project the points on a random plane.
2. For E5, F7, P2Pf, P3P and P4P problems, generate 3 scene points for the validation set. As discussed in Section 3, this is necessary to identify the correct solution, in the case of multiple solutions.
3. Find the ground-truth correspondences \( \bar{x} \) by projecting the scene points to the camera(s) (Table 4).
4. Estimate the true covariance \( C_y \) via MC, by generating 1000 random perturbations to \( \bar{x} \), representing the noise (i.e. \( L = 1000 \)).
5. Estimate the covariance \( \hat{C}_y \) by SUT and FOP, from the input distribution \( N(\bar{x}, C_x) \) (Table 3).
6. Evaluate the quality of the SUT and FOP estimates.

The variables governing the problem generation are reported in Tables 3 and 4. Scene is the dimensions of the volume containing the scene points. The cameras are located at the centre of this region, within the subvolume defined by Camera (Fig. 2). Lens distortion is applied only in the P4P case. In order to evaluate the performance, 3 distinct scene and camera configurations that represent typical real capture conditions are considered:
an indoor studio (Indoor), an outdoor scene viewed by wide-baseline cameras (OWB), and an outdoor scene with narrow-baseline cameras (ONB). The main difference between these scenarios is the camera baseline (i.e., inter-camera distances relative to the size of the scene), which is largest for Indoor, and smallest for ONB. The scene noise is higher for OWB and ONB, to reflect the loss of accuracy in localisation of distant points. The parameters of Indoor are used to generate 3 more datasets, N1, N4, and N9, corresponding to different levels of noise.

The problem generation process described above follows the established practice in the minimal geometry estimation field, and therefore, do not attempt to detect degenerate set-ups explicitly at the problem generation stage [15] [28]. However, if the set-up is

- truly degenerate, which is indicated by the failure of $S(\bar{x})$, or,
- near-degenerate, which is indicated by the failure of $S(.)$ for a member of $\{x\}$, i.e. one of the perturbed versions of $\bar{x}$,

FOP and SUT cannot be computed. In such a case, the problem instance is discarded, and a new one is generated. We choose not to impose an artificial “near-degeneracy prevention” mechanism, like a threshold, beyond the above procedure: if $S(.)$ can be computed at and around $\bar{x}$, even though it may involve a difficult case, we take this sufficient proof that the set-up is valid, and therefore, suitable for covariance estimation. Considering the parameter ranges (Table 3) and the number of experiments, we believe that any pathological cases that might slip through would be too few in number to affect the robust statistics we employ to evaluate the performance (Section 4.1.2).
Table 5: Variation of the median KL divergence with respect to the noise level

<table>
<thead>
<tr>
<th>Case</th>
<th>T2</th>
<th>H4</th>
<th>P6P</th>
<th>F7</th>
<th>F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOP-N1</td>
<td>2.70e-3</td>
<td>1.04e8</td>
<td>1.73e4</td>
<td>8.20e9</td>
<td>6.92e9</td>
</tr>
<tr>
<td>FOP-N4</td>
<td>2.71e-3</td>
<td>2.17e8</td>
<td>2.44e4</td>
<td>1.72e10</td>
<td>1.18e10</td>
</tr>
<tr>
<td>FOP-N9</td>
<td>2.71e-3</td>
<td>2.55e8</td>
<td>3.44e4</td>
<td>2.88e10</td>
<td>1.76e10</td>
</tr>
<tr>
<td>SUT-N1</td>
<td>2.70e-3</td>
<td>4.35e1</td>
<td>1.74</td>
<td>4.40e-1</td>
<td>5.00e-1</td>
</tr>
<tr>
<td>SUT-N4</td>
<td>2.70e-3</td>
<td>1.09e2</td>
<td>3.80</td>
<td>2.11</td>
<td>2.20</td>
</tr>
<tr>
<td>SUT-N9</td>
<td>2.70e-3</td>
<td>1.49e2</td>
<td>6.69</td>
<td>1.08e1</td>
<td>1.24e1</td>
</tr>
</tbody>
</table>

The accuracy of a covariance estimate is quantified by its Kullback-Leibler (KL) divergence from its true value. For the purposes of divergence calculation only, the mean for all estimators is set to $\bar{y} = \bar{S}(\bar{x})$, i.e. the maximum-likelihood estimate of the output. This facilitates a fair comparison of FOP and SUT as covariance estimators. The divergence of the estimated distribution $N(\bar{y}, \hat{C}_y)$ from the true distribution $N(\bar{y}, C_y)$ can be expressed as [20]

$$P = \frac{1}{2} (\text{trace}(\hat{C}_y^{-1}C_y) - \ln \det(\hat{C}_y^{-1}C_y) - D).$$

(9)

where $D$ is the size of the covariance matrix (also, $\dim(y)$ in Table 1). The unit of divergence is nats, which is approximately 1.44 bits.

4.1.2. Results

Fig. 3 presents the performance of FOP and SUT under various noise conditions, as a box-and-whisker diagram of the distribution of $P$. The order statistics depicted in the plots are the 5th, 25th, 50th, 75th and 95th percentiles. The mean is marked with a square, when it is within the range of the y-axis. The results indicate that SUT is superior to FOP for all solvers, with the exception of T2, where a performance parity is observed. Furthermore, SUT yields an accurate estimate of the true covariance, especially in problems with low output dimensionality (Table 1), i.e. T2, E5, P2P, P2Pf and P3P, over the entire noise range.
The poor performance of FOP seems to stem from ill-conditioned covariance estimates, which leads to very large entries in $\hat{C}_y$: the determinant term...
in equation 9 is attenuated by the logarithmic scaling, however, when $\hat{C}_y$ is ill-conditioned, the trace of $\hat{C}_y^{-1}C_y$ tends to reach extreme values. On the other hand, a well-conditioned, but inaccurate covariance estimate is likely to result in a more modest divergence value, which can be observed in most of the low dimensionality problems (up to and including P3P). In contrast, SUT regularly achieves accurate covariance estimates, especially in the $N1$ case. Conditioning is a major issue only in H4. However, the 95th percentile and the mean values suggest that, with the exception of P2P and T2, SUT might occasionally yield ill-conditioned estimates. Still, the range of problem instances that can be successfully handled by SUT is much wider than that of FOP, as the results clearly demonstrate.

For both methods, the quality of the estimated covariance is generally better in problems with lower parameter space dimensionalities (Table 1), and problems involving scene-image correspondences. The former reflects the link between the dimensionality and the complexity of the parameter space: in higher dimensionality problems, the sampling pattern employed by SUT may be less suitable for a good statistical characterisation. The latter might be due to the fact that a coordinate error of 1 pixel and 1 cm are not necessarily equally detrimental to the quality of the output. However, the median value of $P$ for SUT remains relatively small across the entire problem set, whereas FOP achieves similar values only for T2 and P2P.

With the exception of 2-point problems (T2, P2P and P2Pf), the increasing noise variance deteriorates the performance, which is manifested as a larger median divergence, and a wider spread (Fig. 3 and Table 4.1.2). However, it leaves the clear superiority of SUT over FOP intact. The degradation in FOP is less distinguishable, as the already-poor performance is dominated by the aforementioned conditioning issue. As for SUT, within the 25th-75th percentile range, and in lower dimensionality problems, the impact of the noise is small. P4P and P6P suffer a relatively mild degradation, whereas in H4, F8 and F7, higher noise levels imply a significant rise in divergence.

In problems involving 2 cameras (T2, H4 and epipolar geometry estimation), a narrow baseline (i.e. $ONB$ case) accentuates the nonlinearities, and undermines the Gaussianity assumption. Fig. 4 quantifies the associated loss in performance. As expected, the quality of the covariance estimates decreases with the decreasing baseline, i.e. $Indoor$ is the best, and $ONB$ is the worst performance case. However, unlike FOP, SUT proves to be a reliable estimator in $Indoor$ and $OWB$ configurations, and, in certain problems
Figure 4: Box-and-whisker diagrams of the Kullback-Leibler divergence for problems involving image-image correspondences, over various baseline conditions (Indoor, ONB and OWB in Table 3). Note that the y-axis is logarithmic.

(T2 and E5), even in ONB. The impact of baseline on the performance is more pronounced than that of the noise, especially in the fundamental matrix problems, F7 and F8. The only exception is T2, which exhibits a remarkable robustness to decreasing baseline, both in FOP and SUT.

Finally, Fig. 5 depicts the distribution of the divergence for 1-camera problems (P6P and calibration estimation) at different scene configurations, i.e. Indoor and OWB. Despite the higher scene noise, the results are generally similar, with SUT retaining its superiority over FOP. This suggests that the scene size has a limited effect on the performance, and the quality of the covariance estimates is primarily determined by the image noise.

4.2. Real Data Experiments

The performance of FOP and SUT on real data is compared via the camera calibration pipeline presented in [14] (Fig. 6). The algorithm recovers
the unknown internal and external calibration parameters of a moving (principal) camera, which shares a field-of-view with a number of static witness cameras. This is a common set-up in film production, where accurate registration of the principal camera with the witness cameras is essential for many post-production and special effects tasks [14]. The calibration process is as follows:

1. **Given:** Known calibration parameters for the principal camera (*e.g.* focal length for metric calibration or camera position for a nodal motion), full calibration parameters for the witness cameras.

2. Build a sparse 3D scene model from the 2D image feature correspondences observed in the static witness cameras (Fig. 8).

---

**Figure 5:** Box-and-whisker diagrams of the Kullback-Leibler divergence of problems involving scene-image correspondences, over two different scene and camera configurations (*Indoor* and *OWB* in Table 3). Note that the y-axis is logarithmic.
3. For each frame acquired by the moving camera
   (a) Establish scene-image correspondences
   (b) Depending on whether the principal camera is static or moving, or
       the focal length is known, solve the appropriate minimal geometry
       problem to recover the missing calibration parameters (Table 2).
   (c) Feed the resulting calibration measurement to a Kalman filter for
       jitter removal.
   (d) Read the current calibration estimate of the principal camera off
       the state of the Kalman filter.

   The state of the Kalman filter includes the unknown calibration para-
   meters, which, depending on the problem type, may include some or all of
   orientation, position, lens distortion and focal length (Table 2), and their
   rates of change (e.g. rotation, translation, distortion change rate and zoom
   rate). Sample images from the principal camera sequences are presented in
   Fig. 7. The camera set-up, the scene model, and the recovered trajectory for
   Odzemok are illustrated in Fig. 8.

   A Kalman filter requires an indicator of the reliability of each measure-
   ment, in the form of a covariance matrix. The measurement covariance has a
   significant influence on the state vector ($\bar{s}$) and its covariance ($C_s$) through
   the Kalman gain matrix [16]. The real data experiment compares the Gaus-
   sian representing the state, $N(\bar{s}, C_s)$, when the measurement covariance is
   estimated via FOP, SUT and MC. As in the synthetic data experiments,
   the state computed from the MC estimates of the measurement acts as the
Figure 8: The camera set-up and the top-view of the pose trajectory for Odzemok. The trajectory is the line closest to the x-axis.

Figure 9: Box-and-whisker diagrams of the Kullback-Leibler divergence for the camera calibration application. Note that the y-axis is logarithmic.

ground-truth, $N(\bar{s}, \bar{C}_s)$. The Kullback-Leibler divergence of $N(\hat{s}, \hat{C}_s)$, the FOP- or SUT-based state, from the true state is \cite{20}

$$P = \frac{1}{2} \left( \text{trace}(\hat{C}_s^{-1}C_s) + (\hat{s} - \bar{s})^T \hat{C}_s^{-1}(\hat{s} - \bar{s}) - \ln \det(\hat{C}_s^{-1}C_s) - D \right). \quad (10)$$
Fig. 9 illustrates the order statistics of the divergence for FOP and SUT, over the 125-frame sequences used in [14] (the experiments Unicycle1, Unicycle2, Odzemok and Juggler, respectively). The results are in agreement with the synthetic data experiments: SUT outperforms FOP, and yields a very accurate estimate of the state (and its uncertainty) for P2P, P2Pf and P3P. FOP, despite being a clearly inferior alternative, still provides acceptable results for P2P and P2Pf.

5. Conclusion

Like most computer vision algorithms, minimal geometry solvers deal with uncertain inputs, therefore, any output representation without a measure of the associated uncertainty is incomplete. A common representation of the uncertainty is covariance, whose utility is linked to its accuracy. However, the reliability of first-order forward propagation, the most common covariance estimator, is known to suffer in nonlinear problems, which is the case for minimal geometry solvers.

Scaled unscented transformation is a covariance estimator whose superiority to FOP is already established in control and robotics literature [16] [12] for nonlinear transformations. Encouraged by its success in a related field, in this paper, we study the effectiveness of SUT as an alternative to FOP for minimal geometry solvers. To this aim, we first explain how SUT can be implemented for a collection of common solvers, highlight the practical problems for each solver, and propose solutions. Then, we compare SUT and FOP against the estimates of the true covariance by synthetic and real data experiments, through Kullback-Leibler divergence. Both experiments indicate the superiority of SUT over FOP as a covariance estimator, over a number of different noise and scene conditions. Moreover, the covariance estimates of SUT are usually highly accurate and in general, exhibit a better robustness to noise, compared to FOP. Both estimators suffer a severe performance deterioration under narrow baseline conditions (ONB), however, SUT still performs reasonably well for E5 and T2 (Fig. 4). Finally, when employed in Kalman filtering, the use of SUT for computing the measurement covariance matrices yields an accurate estimate of the state and its uncertainty, which is significantly better than that obtained through FOP.

We conclude that, SUT is a better technique than FOP for covariance estimation, offering a considerable performance advantage at effectively no additional computational cost.
Acknowledgments

This work is supported by the TSB projects i3Dlive: interactive 3D methods for live-action media (TP/11/CII/6/I/AJ307D), "SyMMM: Synchronising Multimodal Metadata" (CRD 101033) and the European Commission FP7 project "IMPART: Intelligent Management Platform for Advanced Real-Time Media Processes" (FP7-ICT 316564).

The final publication is available at Elsevier via http://dx.doi.org/10.1016/j.cviu.2014.08.002.

Appendix A. Minimal Geometry Solvers

A detailed discussion of the minimal solvers is beyond the scope of this paper. However, in the interest of self-containedness, we believe a short summary, as provided below, would be helpful.

Appendix A.1. 2-View Triangulation

Consider the camera matrices, \( \mathbf{P}_p \) and \( \mathbf{P}_r \), and the homogeneous 2D points \( \mathbf{p} \) and \( \mathbf{r} \) on their respective image planes. The T2 solver estimates the coordinates of the 3D point, whose projections through the camera matrices yield a pair \( \{ \mathbf{p}'; \mathbf{r}' \} \), closest to \( \{ \mathbf{p}; \mathbf{r} \} \) in the Euclidean sense [11].

\( \mathbf{P}_p \) and \( \mathbf{P}_r \) define the fundamental matrix \( \mathbf{F}_{rp} \). The solver seeks \( \{ \mathbf{p}'; \mathbf{r}' \} \) which exactly satisfies the epipolar constraint \( \mathbf{p}' \mathbf{F}_{rp} \mathbf{r} = 0 \), while minimising the objective function \( \sum \| \mathbf{p}' - \mathbf{p} \|^2 + \| \mathbf{r}' - \mathbf{r} \|^2 \), by solving a 6th degree polynomial [11]. The epipolar constraint ensures that the projection rays linking the camera centres of \( \mathbf{P}_p \) and \( \mathbf{P}_r \) to the points \( \mathbf{p}' \) and \( \mathbf{r}' \) intersect at a 3D point, which is the solution to the 2-view triangulation problem.

Appendix A.2. Epipolar Geometry Estimation

This class of problems involves the estimation of a fundamental or an essential matrix, from a set of 2D image correspondences \( \{ \mathbf{p}_k; \mathbf{r}_k \} \). The solution is a 3x3 matrix \( \mathbf{Y} \), which satisfies the epipolar constraint \( \mathbf{p} \mathbf{Y} \mathbf{r} = 0 \), for \( \{ \mathbf{p}_k; \mathbf{r}_k \} \) [11]. Since \( \mathbf{Y} \) is a rank-2 matrix, \( \det \mathbf{Y} = 0 \) [11]. An essential matrix offers one more constraint: \( \mathbf{Y} \) satisfies the equation \( 2 \mathbf{Y} \mathbf{Y}^T \mathbf{Y} - \text{trace} ( \mathbf{Y} \mathbf{Y}^T ) \mathbf{Y} = 0 \) [19].

For each correspondence, the epipolar constraint provides a relation linear in the elements of \( \mathbf{Y} \). These equations can be stacked into a system of the form \( \mathbf{A} \theta = 0 \), where \( \theta \) is the elements of \( \mathbf{Y} \) reordered into a vector. \( \theta \) is a
linear combination of the basis vectors of $\Theta$, the nullspace of $A$ (with the condition that $\|\theta\| = 1$, which disqualifies the zero-vector as a solution). The solvers differ in their ways to estimate $Y$ from $\Theta$.

**8-point fundamental matrix estimation:** $F_8$ utilises 8 correspondences, therefore $\theta$ is the sole basis vector of $\Theta$. The determinant constraint is enforced post hoc, by removing the component corresponding to the smallest singular value [11].

**7-point fundamental matrix estimation:** In the case of $F_7$, $A$ is a 7x9 matrix, and $\Theta$ has two elements, $\theta_1$ and $\theta_2$. Therefore, the solution is of the form $\theta_1 + \psi\theta_2$. The determinant constraint provides a polynomial equation cubic in $\psi$, whose real roots yield up to 3 solutions [11].

**5-point essential matrix estimation:** $E_5$ requires 5 correspondences, and $A$ has a 4-dimensional nullspace. Therefore, $\theta = \psi_1\theta_1 + \psi_2\theta_2 + \psi_3\theta_3 + \theta_4$. [19] utilises the determinant and the trace constraints to build 10 cubic polynomial equations in $\psi_1, \psi_2$ and $\psi_3$, which are, then, solved via Grobner basis techniques. The real solutions of the polynomial system yield up to 10 essential matrices.

**Appendix A.3. Homography Estimation**

Homography estimation problems deal with the estimation of a matrix $Y$, for a set of correspondences $\{p_k; r_k\}$ satisfying the projective equality relation $p \approx Yr$. This algebraic constraint can be expressed as $p \times Yr = 0$, where $p \times$ is a skew symmetric matrix, determined by the problem type [11] [29]. Each such relation yields two equations linear in the elements of $Y$ which, similar to Appendix A.2, can be stacked into the form $A\theta = 0$. The scale of $\theta$ is fixed by the constraint $\|\theta\| = 1$.

**4-point homography estimation:** Estimation of a 3x3 homography matrix $Y$ relating the homogeneous 2D points requires 4 correspondences. Therefore, $A$ is an 8x9 matrix, nullspace of which is $Y$ [11].

**6-point camera matrix estimation:** A projective camera matrix can be regarded as a 3x4 homography, and can be estimated from 6 2D-3D correspondences. Since 6 correspondences provide 12 equations, $A$ is full-rank, and the solution is the singular vector corresponding to the minimum singular value of $A$ [30].

**Appendix A.4. P2P, P2Pf and P3P Solvers**

A 3D point can be parametrised by the line segment linking it to the camera centre. In the camera reference frame (where the camera centre is at
the origin), for a 2D-3D correspondence \{p; r\}, this line segment is a function of the focal length, \|r\| and the image point p. The mapping between the camera and the world reference frames is an isometry, under which distances between pairs of points are preserved. Therefore, \|r\| can be computed as the distance between the camera centre and r, the corresponding 3D point in the world reference frame. The P2P, P2Pf and P3P solvers leverage this observation, as described below.

**2-point orientation estimation:** The P2P solver [14] estimates the orientation component of a camera matrix from two 2D-3D correspondences, \{p_k; r_k\}, when the camera centre and the intrinsic calibration parameters are available. The known camera centre allows the computation of the distances \|r_k\|, and the known focal length, the direction of the line segments (i.e. the back-projection ray). Therefore, for each r, the corresponding r' in the camera reference frame can be computed. Two such correspondences are sufficient to compute the orientation of the camera [31].

**2-point orientation and focal length estimation:** The P2Pf solver, proposed in [14] estimates the focal length and the orientation of a camera, given a pair of 2D-3D correspondences and the camera centre. The algorithm estimates up to two focal lengths, and then recovers the associated orientations via P2P. In order to set up a constraint on the focal length, first, the cosine of the angle between the projection rays of the correspondences is computed, by applying the cosine law on the triangle defined by the camera centre and the 3D points. This is equal to the dot product of the projection rays, which can be expressed as a quadratic function of the square of the unknown focal length. The constraint can be expressed as a polynomial equation, with up to two positive solutions.

**3-point orientation and position:** The P3P algorithm [10] estimates the orientation and the position of a camera from 3 2D-3D correspondences, given the focal length. Its solution strategy involves locating the 3D points in the camera frame, and then recovering the Euclidean transformation to the world reference frame (via, e.g. [13]). Since the direction of the projection rays can be established from the 2D-3D correspondences, the problem can be cast as the estimation of 3 distances to the camera centre. Observing that the distances between the 3D points are available, application of the cosine law on each triangle involving two of the 3D points and the camera centre yields 3 quadratic constraints. The resulting equation system is solved by computing the roots of a cubic polynomial, which yields up to 3 solutions [10].
Appendix A.5. Simultaneous Estimation of Pose, Focal Length and Lens Distortion

The P4P problem involves the estimation of the pose, focal length and lens distortion of a camera from 4 2D-3D correspondences. [15] utilises the projection equation which links each correspondence $\{p: r\}$ as

$$
\lambda \begin{bmatrix}
    p_1 \\
    p_2 \\
    \mu(p_1^2 + p_2^2)
\end{bmatrix} = Y(K)r,
$$

(A.1)

where $Y(K)$ is the camera calibration matrix defined by the calibration parameters $K$, $\mu$ is the lens distortion coefficient, and $\lambda$ is a projective scaling factor. By a suitable choice of coordinate system, the translation and the focal length parameters can be expressed in terms of the orientation quaternion. $\lambda$ is eliminated by multiplying equation A.1 by a skew symmetric matrix on the left, similar to Appendix A.3. The resulting polynomial equation system is solved via Grobner basis techniques, yielding up to 12 solutions.

Appendix B. Performance Evaluation via an Algebraic Measure

In practice, the computation of Kullback-Leibler divergence (equation 9) could be problematic, as the estimated covariance matrices could be ill-conditioned. If the invertibility of a covariance estimate is of secondary importance, relative Frobenius norm difference can be employed as an alternative, algebraic measure, i.e.

$$
P_{al} = \frac{\|\hat{C}_y - C_y\|_F}{\|C_y\|_F},
$$

(B.1)

where $C_y$ is the estimate of the ground truth (via MC), $\hat{C}_y$ is the covariance estimate (via FOP or SUT), and $\|\cdot\|_F$ is the Frobenius norm operator.

The synthetic data experiments in Section 4.1.2 are repeated with the algebraic error metric. The results, depicted in Fig. B.1, B.2 and B.3, are consistent with those in Section 4.1.2. The only deviations are in P6P, P2Pf and P4P cases, where there appears to be a parity in performance. This can be explained by observing that, as indicated in Sections 3.3 and 3.4, focal length is one of the elements in the minimal parametrisations employed by P6P, P2Pf, and P4P. Also, Table 3 shows that the value of focal length
Figure B.1: Box-and-whisker diagrams of the relative Frobenius norm error over various noise conditions ($N1$, $N4$ and $N9$ in Table 3). Note that the y-axis is logarithmic.

is up to 3 orders of magnitude larger than the orientation and the position parameters. Therefore, the covariance term corresponding to the focal
length dominates equation B.1, and any errors on the remaining entries of \( \hat{C}_y \) become negligible. The experiments for the P2P and P3P cases, whose parametrisations employ only the extrinsic calibration terms, show that the error terms suppressed by the relatively large magnitude of the focal length are, in fact, non-trivial. On the other hand, the inversion operation in equation 9 effectively performs a magnitude normalisation.

Despite the emphasis of equation 9 on invertibility, and of equation B.1 towards fidelity in higher-magnitude entries, taken together, they paint a clear picture: with the exception of the 2-view triangulation case (where FOP performs similarly to SUT), SUT is superior to FOP.

Appendix C. Performance Evaluation against Random Sampling

As discussed in Section 2.2, MC and SUT differ in the way the samples are generated: MC uses a large number number of random samples, whereas
SUT, a smaller number of deterministic samples. In this section, we present experimental results for a variant of MC which employs random sampling, but generates only as many samples as SUT (henceforth, \textit{MC2}).

The experimental setup follows the procedure described in Section 4.1.1. The new experiment involves the generation of \( \{x_k\} \) via Monte-Carlo sampling, however, with \( L = 2M + 1 \), i.e. the same number of samples as SUT. The estimated covariance is evaluated via the Kullback-Leibler divergence and the relative Frobenius norm. The experiment is performed only for the \( N1 \) case, as the performance of \textit{MC2} does not encourage further experiments on other cases, which present more challenging conditions: as Fig. C.1 and C.2 depict, \textit{MC2} is inferior to both \textit{FOP} and \textit{SUT}. 

Figure B.3: Box-and-whisker diagrams of the relative Frobenius norm error of problems involving scene-image correspondences, over two different scene and camera configurations (\textit{Indoor} and \textit{OWB} in Table 3). Note that the y-axis is logarithmic.
A comparison of the sampling strategies of MC, SUT and FOP is helpful to explain the poor performance of MC2. A successful propagation of covariance is possible only if the sample statistics of \( \{x_k\} \) is a good characterisation of the uncertainty of \( x \). As discussed in Section 2.2, MC achieves this by employing a large number of points, whose sample statistics, in the limit, converge to the true covariance, \( C_x \). SUT deterministically selects a small number of points, in a way that matches the first three moments of \( x \). FOP perturbs the measurement vector along each coordinate axis, to generate a symmetric lattice around it. However, MC2 cannot guarantee an accurate characterisation as,

- it employs too few samples, therefore, unlike MC does not offer a representative sampling of the uncertainty volume, and the underlying density function.
- it employs random sampling, which cannot provide the theoretical guarantees enjoyed by SUT, achieved via a careful placement of samples.

The deficiencies of MC2 in sample generation leads to an erroneous \( \hat{C}_y \), and a poor performance.

References


Figure C.1: Box-and-whisker diagrams of the Kullback-Leibler divergence for \( FOP \), \( SUT \) and \( MC2 \), in \( N1 \)


Figure C.2: Box-and-whisker diagrams of the relative Frobenius norm for FOP, SUT and MC2, in $N^1$


