Comparative Analysis of Two-View and Three-View Pose Estimation Algorithms for Image-Based 3D Reconstruction: Fundamental Matrix vs Trifocal Tensor
IACV 2023-2024 Project

Giovanni Versiglioni[1], Luca Magri[2], Federica Arrigoni[2], and Vincenzo Caglioti[2]

[1]**[Student]** giovanni.versiglioni@mail.polimi.it
[2]**[Supervisors]** luca.magri@polimi.it, federica.arrigoni@polimi.it, vincenzo.caglioti@polimi.it

Abstract

Image-based 3D reconstruction is crucial in areas like computer vision and augmented reality. Traditionally, most reconstruction algorithms have focused on using two images at a time. However, using three images together can apply stricter geometric rules, potentially improving accuracy. The trifocal tensor is generally favoured over the fundamental matrix when working with three views. This research project compares existing methods for two-view and three-view pose estimation, challengeing this preference by thoroughly investigating and comparing the performance of the trifocal tensor against the fundamental matrix.

Keywords *Multiple-View Geometry, Pose Estimation, Fundamental Matrix, Trifocal Tensor*

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

FM Fundamental Matrix

EM Essential Matrix

TFT Trifocal Tensor

L-FM Linear Fundamental Matrix Estimation

O-FM Optimized Fundamental Matrix Estimation

L-TFT Linear Trifocal Tensor Estimation

R-TFT Ressl Trifocal Tensor Estimation

N-TFT Nordberg Trifocal Tensor Estimation

FP-TFT Faugeras-Papadopoulo Trifocal Tensor Estimation

PH-TFT Ponce-Hebert Trifocal Tensor Estimation

SVD Singular Value Decomposition

GH Gauss-Helmert Optimization

LM Levenberg-Marquardt Optimization

DLT Direct Linear Transformation

BA Bundle Adjustment

1. Introduction

Since the beginning of computer vision, cameras and images have been key areas of study. Central to this field are challenging tasks like figuring out positions and reconstructing 2D or 3D scenes. These tasks rely on understanding how points in space relate to their images, following the principles of perspective projection in pinhole cameras. This knowledge allows us to triangulate points in space from their projections in images.

Within this framework, the fundamental matrix is a key algebraic tool that captures the relationship between matching points in images. It helps us understand the relative positions and orientations of two camera views, which is crucial for many computer vision applications. Extending this to three views introduces the trifocal tensor, a mathematical construct that represents the relationships among three corresponding image points, known as trilinearities. While it's theoretically possible to create a multi-view matrix for any number of views, practical applications usually focus on pairs or triplets of views. As a result, most multi-view structure-from-motion techniques start with pairs or triplets of images for practical use.

Traditionally, the trifocal tensor is favoured over the fundamental matrix when working with three views. This work challenges this preference by thoroughly investigating and comparing the performance of the trifocal tensor against the fundamental matrix.

1.1. Outline

In Sections (2) and (3) we thoroughly define and explain the fundamental matrix and the trifocal tensor, respectively. Then, in Section (4) we present the methods used to determine camera poses, either using the fundamental matrix or the trifocal tensor. The performances of both are compared with empirical findings in Section (5). These results are analyzed in Section (6), leading us to conclude that while the trifocal tensor has certain advantages, they are not significant enough to definitively consider it superior to the fundamental matrix.

1.2. Notation

In this paper, we adopt specific notation conventions: vectors are denoted by lowercase (v), matrices by uppercase (M), tensors by calligraphic bold uppercase ($\boldsymbol{\mathcal{T}}$), and tensors' correlation slices (*i.e.*, matrices) by bold uppercase ($\boldsymbol{T}_i$).

The 3×3 matrix representation of the cross product with a 3-vector v is indicated by $[v]_\times w$, *i.e.*, $[v]_\times w = v\times w$, where w represents any given vector.

The L^2 norm of a vector v is denoted as $\|v\|$, while for matrices or tensors, it represents the L^2 norm of the vector constructed from their coefficients. The Frobenius norm of a matrix M is denoted as $\|M\|$, while for a tensor $\boldsymbol{\mathcal{T}}$, it signifies the square root of the sum of squares of all its elements, denoted as $\|\boldsymbol{\mathcal{T}}\| := \sqrt{\sum_{i,j,k}(\boldsymbol{T}_i^{jk})^2}$.

Additionally, $|M|$ refers to the determinant of matrix M.

2. The Fundamental Matrix

In this section, we begin by defining the fundamental matrix. Next, we outline numerical methods for estimating the fundamental matrix using point correspondences between two images. We start by using linear equations from epipolar constraints to build a basic framework. Then, we delve into Gauss-Helmert Optimization (GH) [7] to improve precision and robustness in our analysis. A similar approach will be carried out for the trifocal tensor later on (Section 3).

2.1. Definition

The Fundamental Matrix (FM) is defined by the equation

$$x'^{\top} F x = 0 \tag{2.1}$$

for any pair of matching points $x \leftrightarrow x'$ in two images.

2.2. Linear Computation

Given sufficiently many point matches (*i.e.*, at least 7), Equation (2.1) can be used to compute the unknown matrix F. In particular, each point match gives rise to one linear equation in the unknown entries of F. Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0. \tag{2.2}$$

From a set of n point matches, we derive the set of linear equations

$$Af = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} f = 0, \tag{2.3}$$

where f is the 9-vector made up of the entries of F in row-major order.

The 8-point algorithm stands as the most straightforward approach for computing the FM. This involves constructing and solving a set of linear equations, typically using the least squares method. The original algorithm is due to [6].

Algorithm 1: Normalized Eight Point Algorithm (L-FM)

Objective: Given $n \geq 8$ image point correspondences $\{x_i \leftrightarrow x'_i\}$, determine the FM F such that ${x'}_i^{\top} F x_i = 0$.

Algorithm:

(i) **Normalization:** Transform the image coordinates according to $\hat{x}_i = Tx_i$ and $\hat{x}'_i = Tx'_i$, where T and T' are normalizing transformations consisting of a translation and a scaling.

(ii) Find the FM $\hat{F}'$ corresponding to the matches $\{x_i \leftrightarrow x'_i\}$ by

  (a) **Linear solution:** Determine $\hat{F}$ from the singular vector corresponding to the smallest singular value of $\hat{A}$, where $\hat{A}$ is composed from the matches $\{x_i \leftrightarrow x'_i\}$ as defined in Equation (2.3).

  (b) **Constraint enforcement:** Replace $\hat{F}$ by $\hat{F}'$ such that $|\hat{F}'| = 0$ using the SVD.

(iii) **Denormalization:** Set $F = T'^{\top} \hat{F}' T$. Matrix F is the FM corresponding to the original data $\{x_i \leftrightarrow x'_i\}$.

2.3. Optimized Computation

Algorithm 2: GH Algorithm for the FM (O-FM)

Objective: Given $n \geq 8$ image point correspondences $\{x_i \leftrightarrow x_i'\}$, determine the FM F such that ${x'}_i^\top F x_i = 0$.

Algorithm:

(i) **Initial Linear Estimation:** Algorithm (1).

(ii) **Optimization:** Apply GH to iteratively reduce the estimation error.

3. The Trifocal Tensor

3.1. Derivation and Definition

In this section, we explore the trifocal tensor by examining the relationships among three corresponding lines. When a 3D line is viewed from three different perspectives, it creates constraints on the image lines seen in each view. Geometrically, the planes formed by back-projecting these lines from each view must intersect along the same 3D line, which projects onto the matched lines in the images. These geometric constraints can then be expressed algebraically.

We examine a set of corresponding lines denoted as $l \leftrightarrow l' \leftrightarrow l''$, alongside canonical camera matrices for the three views: $P = [I|0]$, $P' = [A|a_4]$, and $P'' = [B|b_4]$, where A and B are 3×3 matrices, and a_i and b_i represent the columns of their respective camera matrices. The epipoles a_4 and b_4 in views two and three, derived from the first camera, are denoted as e' and e'', respectively, with $e' = P'C$ and $e'' = P''C$, where C is the first camera center.

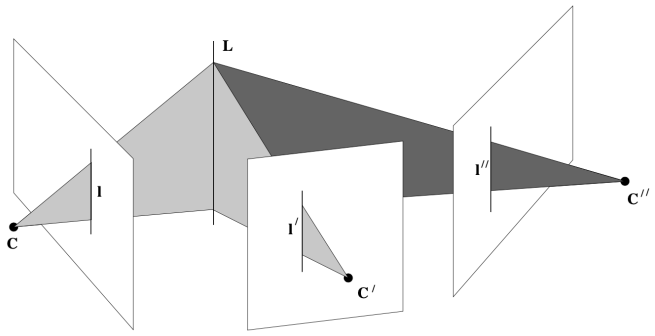


Figure 1. A line L in 3-space is imaged as the corresponding triplet $l \leftrightarrow l' \leftrightarrow l''$ in three views indicated by their centres, C, C' , C'' , and image planes. Conversely, corresponding lines back-projected from the first, second and third images all intersect in a single 3D line in space. [3]

Considering projective transformations, we focus on properties such as image coordinates and 3D incidence relations, which remain invariant. Each image line is projected back to a plane, with these planes constrained to intersect at the common line in 3D space. This constraint is algebraically expressed by ensuring that a specific matrix $M = [\pi, \pi', \pi'']$ has a rank of 2. Here, π, π', and π'' represent the back-projected planes of the image lines in each view

$$\pi = P^\top l = \begin{pmatrix} l \\ 0 \end{pmatrix} \quad \pi' = P'^\top l' = \begin{pmatrix} A^\top l' \\ a_4^\top l' \end{pmatrix} \quad \pi'' = P''^\top l'' = \begin{pmatrix} B^\top l'' \\ b_4^\top l'' \end{pmatrix}. \tag{3.1}$$

The latter intersection constraint induces the following incidence relation amongst the image lines

$$l_i = l'^{\top} \boldsymbol{T}_i l'', \tag{3.2}$$

where $\boldsymbol{T}_i = a_i b_4^{\top} - a_4 b_i^{\top}$, $i = 1, 2, 3$. The set of three matrices $[\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3]$ constitute the Trifocal Tensor (TFT) in matrix notation. Hence, the incidence relation (3.2) can be expressed as

$$l^{\top} = l'^{\top} [\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3] l''. \tag{3.3}$$

3.2. Tensor Notation

Image points and lines are represented by homogeneous column and row 3-vectors, respectively. The ij-th entry of a matrix A is denoted by a_i^j, index i being the contravariant (row) index and j being the covariant (column) index. If the canonical 3×4 camera matrices are $P = [I|0]$, $P' = [a_j^i]$, and $P'' = [b_j^i]$, the definition of the TFT in tensor notation becomes

$$\boldsymbol{\mathcal{T}}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k. \tag{3.4}$$

The placement of indices in the tensor (two contravariant and one covariant) follows the arrangement of indices on the right side of the equation. Hence, the trifocal tensor is a mixed contravariant-covariant valency 3 tensor denoted by a homogeneous array of size $3 \times 3 \times 3$ (*i.e.*, 27 elements) and possesses 18 degrees of freedom.

Thus, the fundamental incidence relation (3.2) is expressed as

$$l_i = l'_j l''_k \boldsymbol{\mathcal{T}}_i^{jk}. \tag{3.5}$$

3.3. Trilinearities

Similarly to the fundamental matrix in two-view geometry, the trifocal tensor encodes relationships between points and lines across three perspectives. These relationships are denoted as trilinearities: "tri" since every monomial in the relation involves a coordinate from each of the three image elements involved, and linear because the relations are linear in each of the algebraic entities (*i.e.*, the three arguments of the tensor). The following equation portrays a point-point-point (P-P-P) correspondence

$$[x']_{\times} \left(\sum_i x^i \boldsymbol{T}_i \right) [x'']_{\times} = 0_{3\times 3}, \tag{3.6}$$

with x, x', and x'' being the homogeneous coordinates of corresponding points in three images. However, other trilinear relations can be derived, such as L-L-L, P-L-L, P-L-P, P-P-L, and P-P-P, where P stands for point and L stands for line. These trilinearities are invariant under projective transformations, ensuring robustness across different camera configurations and scenes.

Among the nine scalar equations in (3.6), only four are linearly independent. They manifest linearity with respect to the parameters of the trifocal tensor and trilinearity with respect to the image coordinates. When viewed in pairs, the incidence relationships established by the FMs for the corresponding triplet x, x', and x'' consist of a group of three equations that are linear with respect to the parameters of the FMs and bilinear with respect to the image points

$$x'^{\top} F_{21} x = 0 \quad x''^{\top} F_{31} x = 0 \quad x''^{\top} F_{32} x' = 0, \tag{3.7}$$

where the involved FMs are

$$F_{21} = [a_4]_{\times} A \quad F_{31} = [b_4]_{\times} B \quad F_{32} = [b_4 - BA^{-1} a_4]_{\times} BA^{-1}. \tag{3.8}$$

3.4. Linear Computation

The TFT can be derived from a linear system described by the trilinear relationships outlined in Equation (3.6). Each triplet yields nine equations that are linear with respect to the tensor's parameters, yet only four of these equations are linearly independent. To solve this linear system, a minimum of seven correspondences is required, with the additional constraint $||\mathcal{T}|| = 1$. If more triplets are available, a solution minimizing the algebraic error can be obtained via SVD. However, the resulting TFT may not always be valid.

To fix this, a valid TFT can be computed through an algebraic minimization algorithm that parallels the linear process employed to find the FM.

Algorithm 3: Algebraic Minimization Algorithm (L-TFT)

Objective: Given a set of point and line correspondences in three views, compute the TFT.
Algorithm:

(i) From the set of point and line correspondences compute the set of equations of the form $At = 0$, where t is the 27-vector made up of the entries of the TFT.

(ii) Solve these equations using the least-squares solution to constrained systems, in order to find an initial estimate of $\mathcal{T}_i^{jk}$.

(iii) Find the two epipoles e' and e'' from $\mathcal{T}_i^{jk}$ as the common perpendicular to the left null-vectors of the three slices $\boldsymbol{T}_i$.

(iv) Construct the 27×18 matrix E such that $t = Ea$, where a is the vector representing entries of a_i^j and b_i^k, and where E espresses the linear relationship $\mathcal{T}_i^{jk} = a_i^j e''^k - e'^j b_i^k$.

(v) Minimize $||AEa||$ subject to $||Ea|| = 1$. Compute the error vector $\epsilon = AEa$.

(vi) **Iteration:** The mapping $(e', e'') \mapsto \epsilon$ is a mapping from $\mathbb{R}^6$ to $\mathbb{R}^{27}$. Iterate on the last two steps with varying e' and e'' using the Levenberg-Marquardt Optimization (LM) algorithm to find the optimal e', e'' pair. Hence find the optimal $t = Ea$ containing the entries $\mathcal{T}_i^{jk}$.

3.5. Optimized Computation

Several concise and consistent descriptions of the TFT have been suggested in prior literature [1, 2, 4, 8, 9, 10, 11, 13], where the term consistent ensures that the tensor does satisfy its constraints, thus is geometrically valid. We've opted to concentrate on four representative ones that can be seamlessly integrated into the pose estimation procedure, exploiting GH.

Algorithm 4: GH Algorithm for the TFT (R-TFT, N-TFT, FP-TFT, PH-TFT)

Objective: Given a set of point and line correspondences in three views, compute the TFT.
Algorithm:

(i) **Initial Linear Estimation:** Algorithm (3).

(ii) **Optimization:** Apply GH with respect to one of the parametrizations to iteratively reduce the estimation error.

Ressl (R-TFT) Ressl, in his thesis [11], introduced a minimal parameterization relying on algebraic constraints within correlation slices of the TFT. This formulation consists of 20 parameters and 2 constraints, and expresses the three slices T_i as follows

$$\boldsymbol{T}_i = [s_i, vs_i + m_i e_{31}, ws_i + n_i e_{31}]^\top, \tag{3.9}$$

where $s_i \in \mathbb{R}^3$ are such that $||(s_1 s_2 s_3)|| = 1$, $e_{31} \in \mathbb{R}$ with $||e_{31}|| = 1$, and $v, w, m_i, n_i \in \mathbb{R}$.

This parameterization directly links to the epipoles: where $e_{21} = a_4$ is proportionate to $(1, v, w)^\top$ and signifies the epipole of the second view, *i.e.*, the projection of the first camera centre onto the second image, and $e_{31} = b_4$ signifies the epipole of the third view, *i.e.*, the projection of the first camera centre onto the third image.

Nordberg (N-TFT) Another approach to parameterize the TFT involves three 3×3 orthogonal matrices: U, V, W, as mentioned in [8]. These matrices transform the original tensor into a sparse form, denoted as $\widetilde{\boldsymbol{\mathcal{T}}}$

$$\widetilde{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}}(U \otimes V \otimes W), \tag{3.10}$$

containing only 10 non-zero parameters, up to scale) where the tensor operation $\otimes$ corresponds to the matrix operation on the slices $\widetilde{\boldsymbol{T}}_{\boldsymbol{i}} = V^\top(\sum_m U_{m,i}\boldsymbol{T}_{\boldsymbol{m}})W$. The scale can be fixed by imposing $||\widetilde{\boldsymbol{\mathcal{T}}}|| = 1$.

For canonical cameras, such orthogonal matrices can be computed as

$$\begin{aligned}
U_0 &= (A^{-1}a_4, [A^{-1}a_4]_\times{}^2 B^{-1}b_4, [A^{-1}a_4]_\times B^{-1}b_4) \\
U &= U_0(U_0^\top U_0)^{-\frac{1}{2}} \\
V_0 &= (a_4, [a_4]_\times AB^{-1}b_4, [a_4]_\times{}^2 AB^{-1}b_4) \\
V &= V_0(V_0^\top V_0)^{-\frac{1}{2}} \\
W_0 &= (b_4, [b_4]_\times BA{-}1a_4, [b_4]_\times{}^2 BA^{-1}a_4) \\
W &= W_0(W_0^\top W_0)^{-\frac{1}{2}},
\end{aligned} \tag{3.11}$$

and each matrix can be represented by 3 parameters, resulting in a total of 19 parameters for $\boldsymbol{\mathcal{T}}$, along with one constraint to determine the scale of $\widetilde{\boldsymbol{\mathcal{T}}}$.

However, a notable drawback of this particular parameterization arises when the camera centers are collinear. In such cases, the matrices U_0, V_0, and W_0 become singular, rendering it impossible to compute orthogonal matrices U, V, W from them. Consequently, this parameterization is only applicable when the camera centers are non-collinear.

Faugeras and Papadopoulo (FP-TFT) The work outlined in [9] introduces a set of 12 algebraic equations, serving as constraints to fully define a TFT. These include three constraints of third-degree, corresponding to the slices' determinants being zero, $|\boldsymbol{T}_{\boldsymbol{i}}| = 0$ for $i \in \{1, 2, 3\}$, and an additional nine constraints of sixth-degree. These sixth-degree constraints involve combinations of various determinants of the tensor's elements

$$\begin{aligned}
&|\boldsymbol{T}_{\cdot}^{j_1 k_1}\boldsymbol{T}_{\cdot}^{j_1 k_2}\boldsymbol{T}_{\cdot}^{j_2 k_2}||\boldsymbol{T}_{\cdot}^{j_1 k_1}\boldsymbol{T}_{\cdot}^{j_2 k_1}\boldsymbol{T}_{\cdot}^{j_2 k_2}| - \\
&|\boldsymbol{T}_{\cdot}^{j_2 k_1}\boldsymbol{T}_{\cdot}^{j_1 k_2}\boldsymbol{T}_{\cdot}^{j_2 k_2}||\boldsymbol{T}_{\cdot}^{j_1 k_1}\boldsymbol{T}_{\cdot}^{j_2 k_2}\boldsymbol{T}_{\cdot}^{j_1 k_2}| = 0,
\end{aligned} \tag{3.12}$$

where $j_1, j_2, k_1, k_2 \in \{1, 2, 3\}$ with $j_1 \neq j_2$ and $k_1 \neq k_2$, and where $\boldsymbol{T}_{\cdot}^{jk}$ represents the vector $(\boldsymbol{T}_1^{jk}, \boldsymbol{T}_2^{jk}, \boldsymbol{T}_3^{jk})^\top$.

This collection isn't minimal because it requires just 9 constraints to fully define a valid tensor.

Ponce and Hebert Π Matrices (PH-TFT) An alternative method of characterizing the 3-view model has been investigated in [10]. By analyzing how three lines intersect in space, a trio of matrices has been derived, each associated with principal lines. These matrices, comprising a total of 27 parameters, impose constraints on the correspondence between three image points. For a configuration involving three cameras with non-collinear centers and three image points x_1, x_2, x_3, there exist three 4×3 matrices, denoted as $\Pi_i = (\pi_{1i}, \pi_{2i}, \pi_{3i}, \pi_{4i})^\top$, each scalable, where $\pi_{ii} = (0, 0, 0)^\top$, and they satisfy

$$x_1^\top(\pi_{41}\pi_{32}\top - \pi_{31}\pi_{42}\top)x_2 = 0 \tag{3.13}$$

$$x_1^\top(\pi_{41}\pi_{23}\top - \pi_{21}\pi_{43}\top)x_3 = 0 \tag{3.14}$$

$$x_3^\top(\pi_{43}\pi_{13}\top - \pi_{12}\pi_{43}\top)x_3 = 0 \tag{3.15}$$

$$(\pi_{21}^\top x_1)(\pi_{32}^\top x_2)(\pi_{13}^\top x_3) = (\pi_{31}^\top x_1)(\pi_{12}^\top x_2)(\pi_{23}^\top x_3), \tag{3.16}$$

if, and only if, the x_i form a triplet of corresponding points.
Ponce and Hebert propose the 6 following homogeneous constraints

$$\begin{gathered}\pi_{21}^1 = \pi_{32}^2 = \pi_{13}^3 =, 0 \\ \pi_{31}^2 = \pi_{41}^3, \quad \pi_{12}^3 = \pi_{42}^1, \quad \pi_{23}^1 = \pi_{43}^2.\end{gathered} \tag{3.17}$$

This can be accomplished through a projective transformation of the space, reducing the parameters to 21. By imposing three norm constraints on the matrices, $||\Pi_i|| = 1$, we achieve the most concise representation.

Similar to the trilinearities (3.6), these parameters yield four equations detailing the incidence relation for image points. Equations (3.13), (3.14), and (3.15) are bilinear regarding the points and are entirely analogous to the epipolar equations provided by the FMs. Equation (3.16), however, is trilinear in the image points, and this allows to characterise the correspondence of three points when one lays on the line connecting two epipoles (while FMs are not able to do it). This highlights the geometric significance of leveraging three views instead of individual pairs in characterizing matches.

A primary limitation of the Π matrices is their exclusive applicability to non-collinear camera centers, just like Nordberg's parametrization.

4. Pose Estimation

We can derive the epipoles, which are the projections of the first camera centre onto the second and third images, from a TFT $\boldsymbol{\mathcal{T}}$. The epipole e_{21} is found as the common point of intersection among the lines represented by the left null-vectors of $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, and $\boldsymbol{T}_3$. Similarly, the epipole e_{31} is determined as the shared point of intersection among the lines represented by the right null-vectors of $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, and $\boldsymbol{T}_3$. Subsequently, we can compute the FMs as

$$\begin{aligned}F_{21} &= [e_{21}]_\times[\boldsymbol{T}_1 e_{31}, \boldsymbol{T}_2 e_{31}, \boldsymbol{T}_3 e_{31}] \\ F_{31} &= [e_{31}]_\times[\boldsymbol{T}_1^\top e_{21}, \boldsymbol{T}_2^\top e_{21}, \boldsymbol{T}_3^\top e_{21}].\end{aligned} \tag{4.1}$$

The Essential Matrix (EM) is the specialisation of the FM to the case of normalized image coordinates. The EMs here can be derived from F_{21}, F_{31}, and the calibration matrices K_i, using the formula $[t_{ij}]_\times R_{ij} = E_{ij} = K_i^\top F_{ij} K_j$. From these EMs, the relative orientations (R_{21}, t_{21}) and (R_{31}, t_{31}) can be extracted through the SVD of E_{21} and E_{31}, with each translation vector's scale remaining unknown. To establish an overall scale, we set

$\|t_{21}\| = 1$, while the relative scale λ of t_{31} can be determined by triangulating the space points $\{X^n\}_n$ from the first two cameras' projections and minimizing the algebraic error relative to the third image, as shown

$$\arg\min_{\lambda\in\mathbb{R}} \sum_{n=1}^{N} \left\| x_3^n \times \left(K_3 \left(R_{31} X^n + \lambda \frac{t_{31}}{\|t_{31}\|} \right) \right) \right\|. \tag{4.2}$$

The latter admits a closed form solution. So, either from the TFT or the FMs, we possess a method for computing the camera poses.

4.1. Bundle Adjustment

In pose estimation, a frequent final stage involves refining the orientations through Bundle Adjustment (BA). This process aims to minimize the square reprojection error across potential camera orientations and spatial points. For N correspondences and M = 3 cameras

$$\min_{\{R_j,t_j\}_j,\{X^i\}_i} \epsilon^2, \quad \epsilon^2 = \sum_{i=1}^{N}\sum_{j=1}^{M} d\left(x_j^i, K_j(R_j X^i + t_j)\right)^2, \tag{4.3}$$

where x_j^i is for the homogeneous coordinates of the observed image point, and the distance d is the Euclidean distance in homogeneous coordinates

$$d\left((x,y,z)^\top, (t,u,v)^\top\right)^2 = \left(\frac{x}{z} - \frac{t}{v}\right)^2 + \left(\frac{y}{z} - \frac{u}{v}\right)^2. \tag{4.4}$$

The optimization procedure can be executed using the LM algorithm [5].

Algorithm 5: Pose Estimation Algorithm

Objective: Given FM or TFT, extract camera poses.

Algorithm:

(i) If employing TFT, derive the epipoles e_{21}, e_{31} first, and then compute the fundamental matrices F_{21}, F_{31} as stated in Equation (4.1); otherwise (employing FM) go to step (ii).

(ii) Compute essential matrices E_{21}, E_{31} from the fundamental matrices F_{21}, F_{31} and the calibration matrices K_i.

(iii) Exploit essential matrices to determine rotations R_2, R_3 and translations t_2, t_3.

(iv) Apply BA to refine rotations and translations (*i.e.*, orientations), by minimizing the squared reprojection error as stated in Equation (4.3).

(v) Triangulate 3D points from their image projections using the Direct Linear Transformation (DLT) algorithm, obtaining the reconstructed 3D scene.

---

5. Experiments

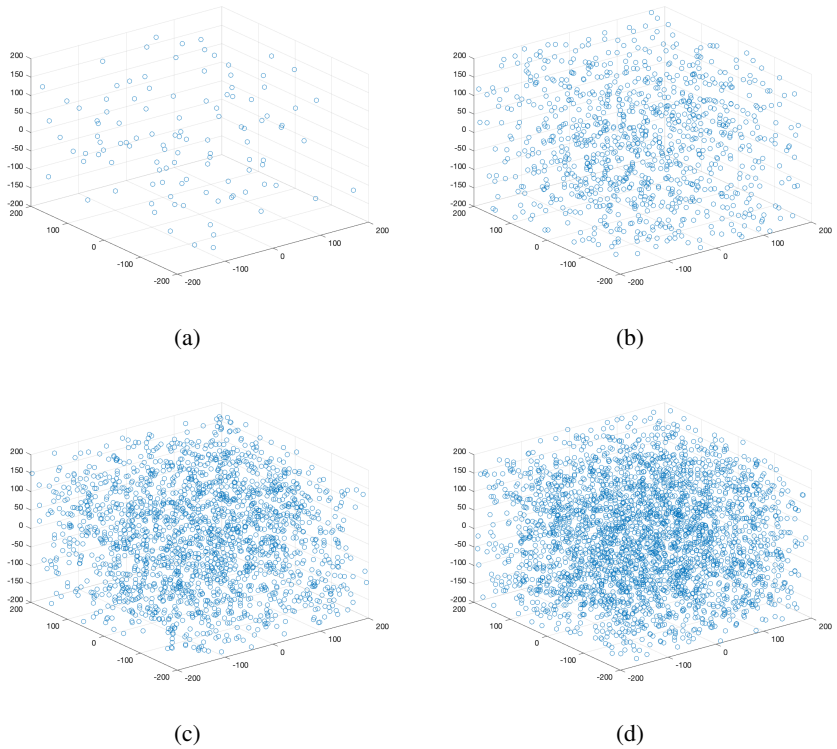
We put into action and assessed the outcomes of pose estimation for both synthetic and real data employing both the fundamental matrix and the trifocal tensor. [1]

As for the FM, we compute it both linearly (L-FM) and through GH (O-FM), whereas for the TFT we employ linear computation (L-TFT) and apply GH using minimal parametrizations proposed by Ressl (R-TFT), Nordberg (N-TFT), Faugeras & Papadopoulo (FP-TFT), and Ponce & Hebert (PH-TFT).

Additionally, we present the result obtained through BA, which is initialised using any of the methods mentioned above. Remarkably, our experiments reveal that all initialisations yield nearly identical final poses post-minimization with BA in the majority of cases..

5.1. Synthetic Data

We conducted trials on synthetic data to assess pose estimation using both the FM and the TFT across various configurations. The standard experimental setup consists of a collection of spatial points situated within a 400mm-sided cube centred at the world's origin, as shown in Figure (2). Points are projected onto three views, and Gaussian noise with a standard deviation of 1 pixel is applied to the image points, unless specified otherwise. The image dimensions are 1800×1200 pixels, and since we assume a fixed focal length of 50mm, we represent a sensor size of $36mm \times 24mm$. All cameras are aligned to focus on the origin. Results are averaged over 30 simulations of data. [2]



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

Figure 2. Set of 3D points considered in the synthetic experiment setup: (a) 100 points simulation, (b) 1000 points simulation, (c) 2000 points simulation, (d) 3000 points simulation.

[1]The MATLAB code to reproduce these experiments is available at the GitHub repository: https://github.com/versi379/Two-View-Three-View-Pose-Estimation.git.

[2]Synthetic trials are developed in the MATLAB script *SyntheticExperiments.m* of the GitHub repository.

Metrics varying Noise

Metrics before and after BA, against Gaussian noise level added to the data points, are shown respectively in Figure (3) and Figure (4).

Initially, all methods show increasing reprojection, rotation, and translation errors as noise rises, with FP-TFT and O-FM demonstrating the lowest errors, indicating higher accuracy. Conversely, L-TFT and R-TFT exhibit higher errors, suggesting less robustness to noise. The number of iterations generally increases with noise, with N-TFT requiring the most iterations, and L-TFT showing the fastest computation times.

However, after BA, all methods display significantly reduced errors and a linear increase with noise, indicating improved precision. The number of iterations and computational times remain relatively stable post-adjustment. Overall, these plots highlight a trade-off between accuracy and computational efficiency, with BA enhancing performance across all methods, making them more precise and robust to noise.

Metrics varying Focal Length

Metrics before and after BA, against varying focal length, are shown respectively in Figure (5) and Figure (6).

Before BA, the TFT methods generally showed higher errors than the FM methods. The L-TFT had the highest initial errors in reprojection, rotation, and translation, while both the L-FM and O-FM performed better in initial estimates.

After applying BA, the differences between parametrizations became less pronounced, especially for reprojection and rotation errors. The FP-TFT and PH-TFT methods seemed to perform consistently well across different metrics. The L-TFT, while greatly improved, still showed higher errors and computation time in some cases compared to other parametrizations.

Metrics varying Number of Points

Metrics before and after BA, against the number of points considered in the synthetic scene, are shown respectively in Figure (7) and Figure (8).

Pre-BA performances showed errors (reprojection, rotation, translation) being significantly higher for all methods, especially with a small number of points. The L-TFT method generally showed the highest initial errors, whereas both the FM methods often performed better in initial estimates.

Post-BA performances showed error significantly reduced, with reprojection error decreasing from hundreds to less than 0.1 pixels, and rotation & translation errors reduced to near-zero.

Metrics varying Camera Angle

Metrics before and after Bundle Adjustment, against the varying angle among the three camera centers, are shown respectively in Figure (9) and Figure (10).

Again, results after BA are clearly more accurate with respect to the initial ones, with errors being almost null for all methods except N-TFT, which showed peaks for certain angular values.

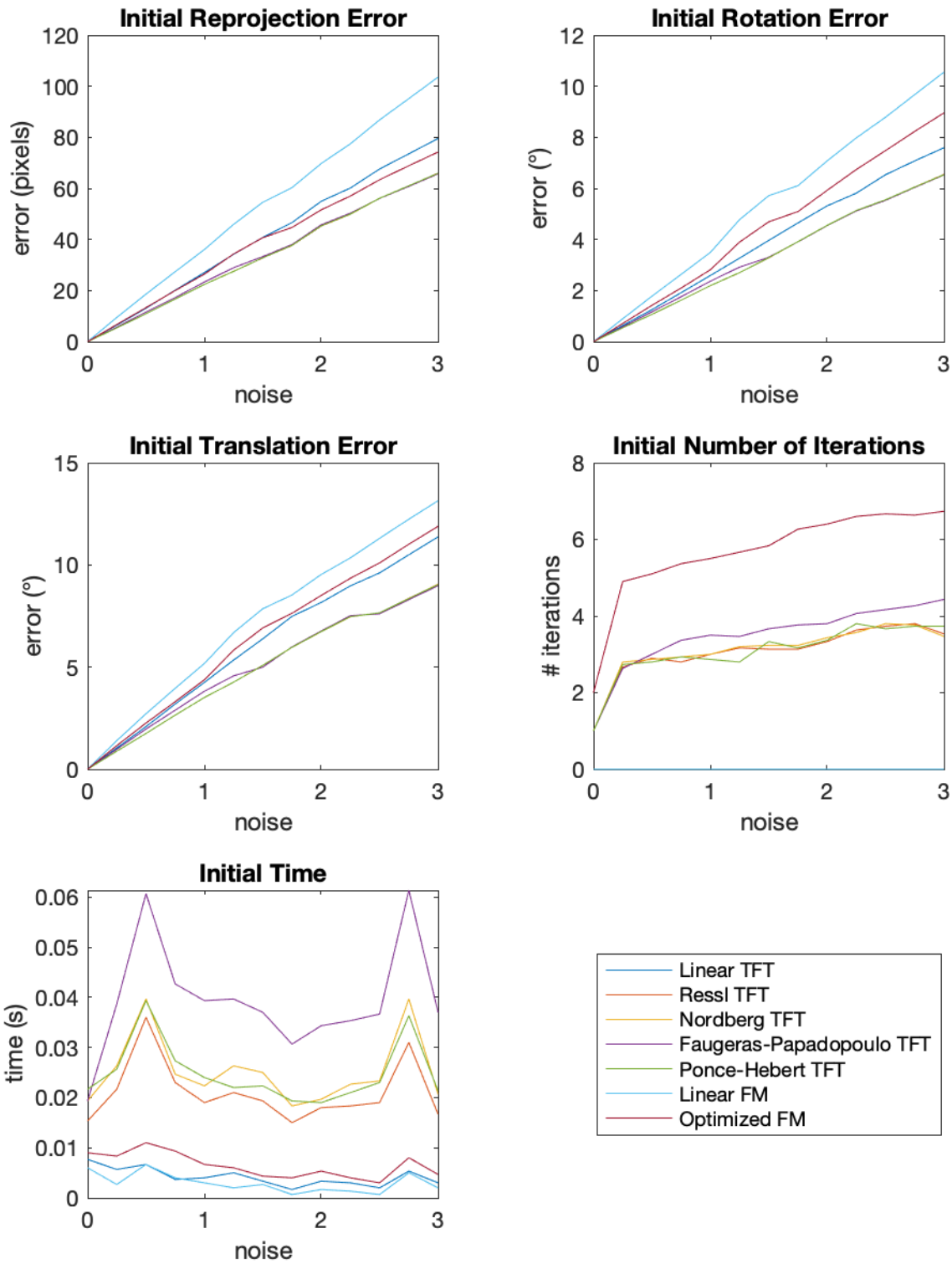


Figure 3. Initial reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left); when varying the Gaussian noise added to the image points.

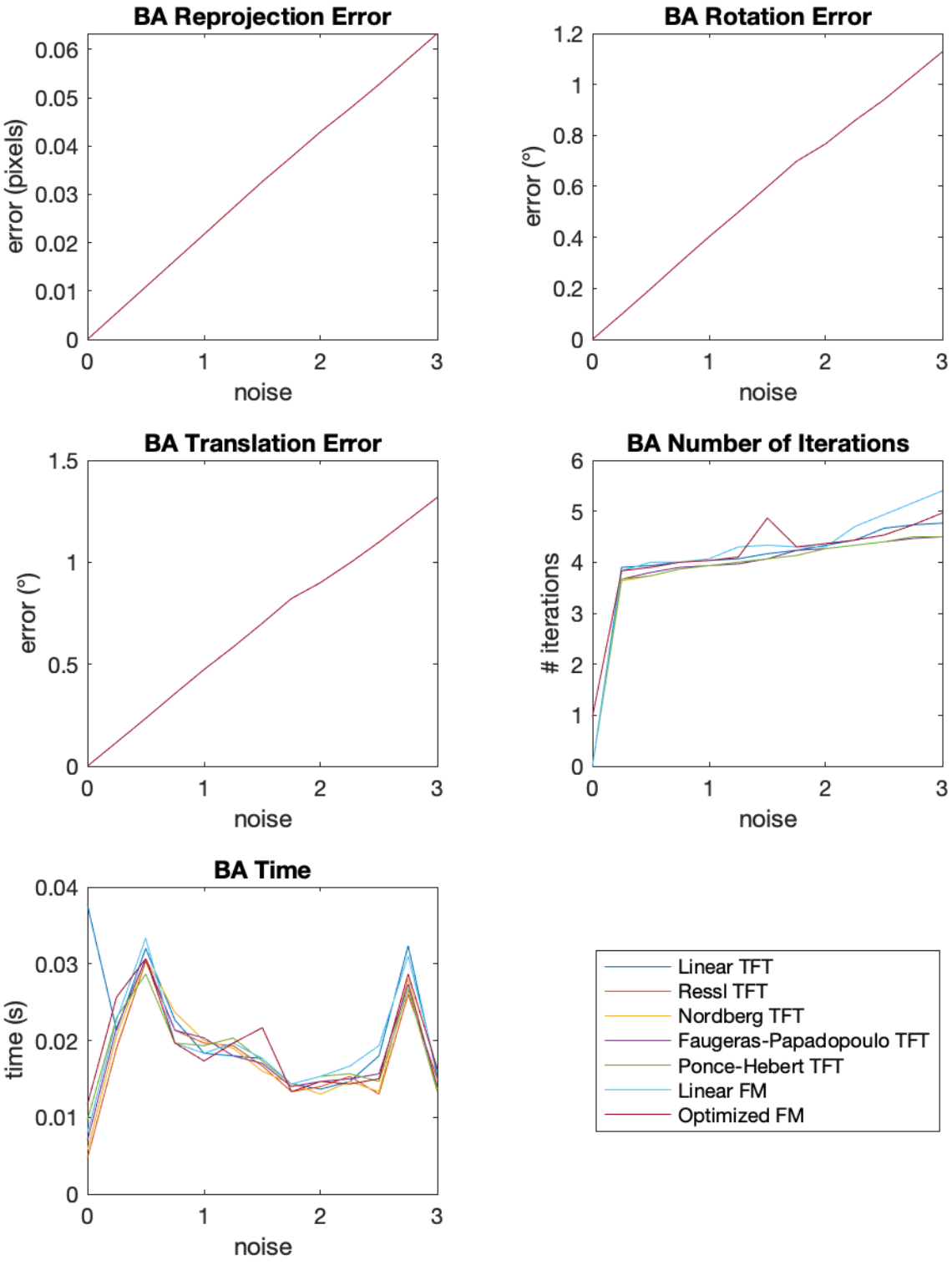


Figure 4. Reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left) after BA; when varying the Gaussian noise added to the image points.

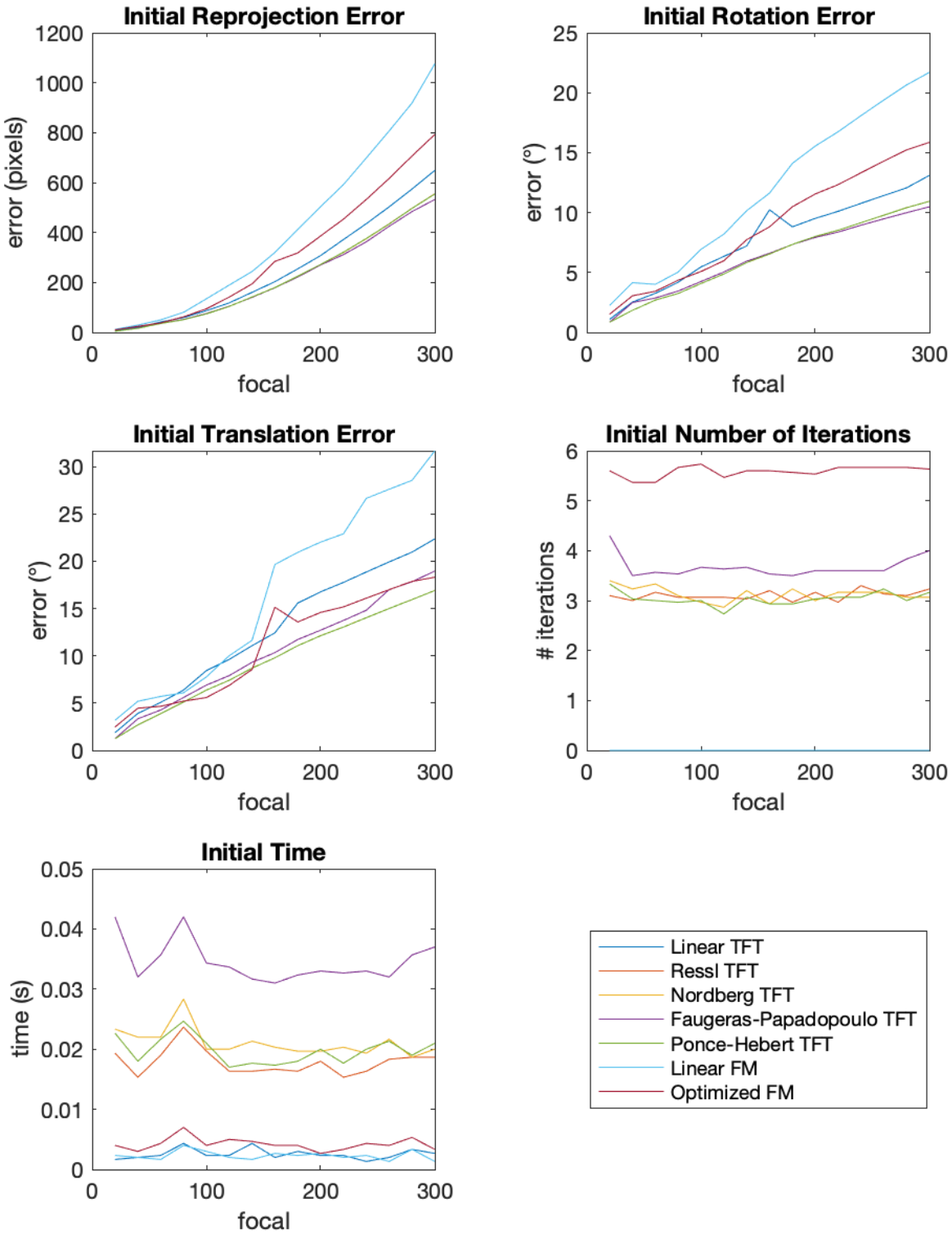


Figure 5. Initial reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left); when varying the focal length.

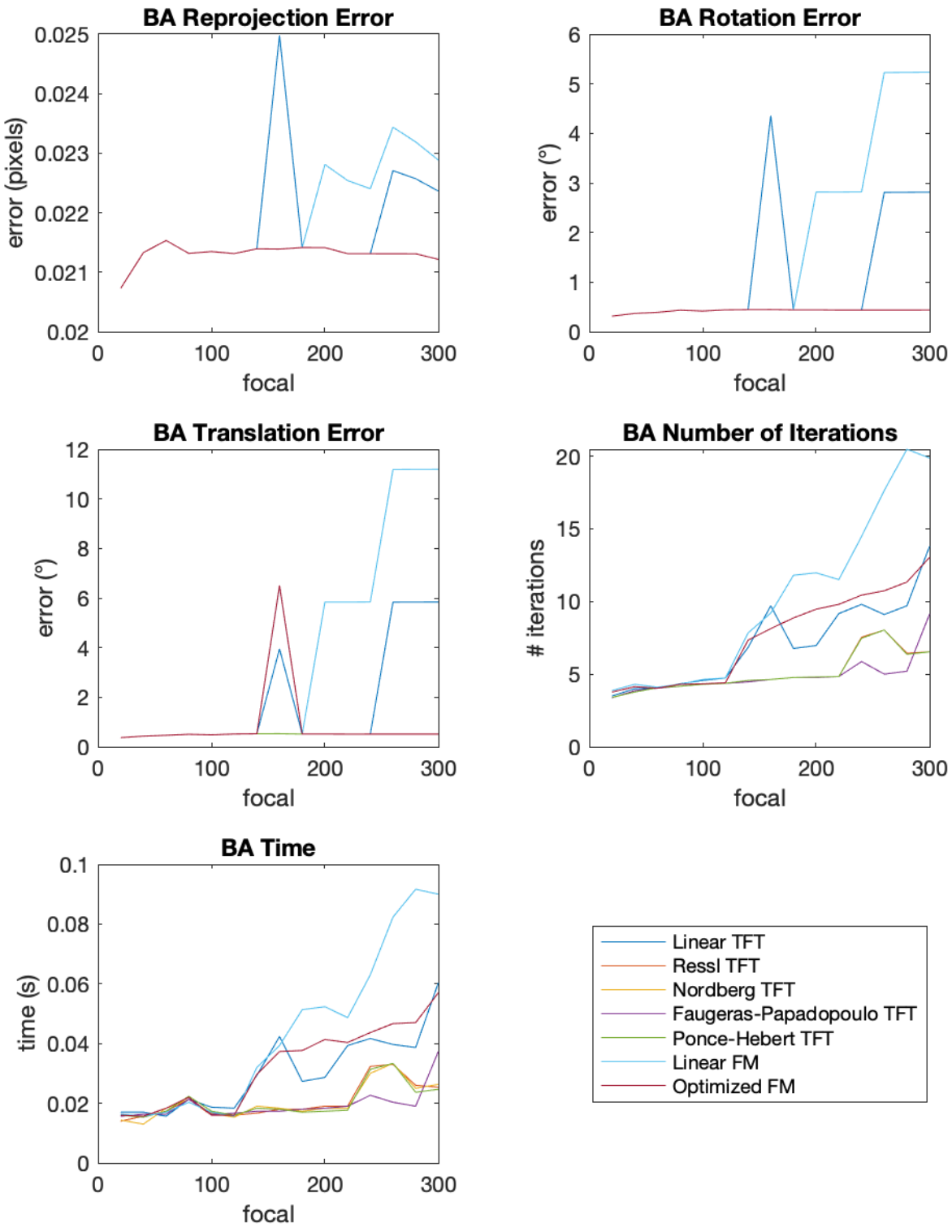


Figure 6. Reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left) after BA; when varying the focal length.

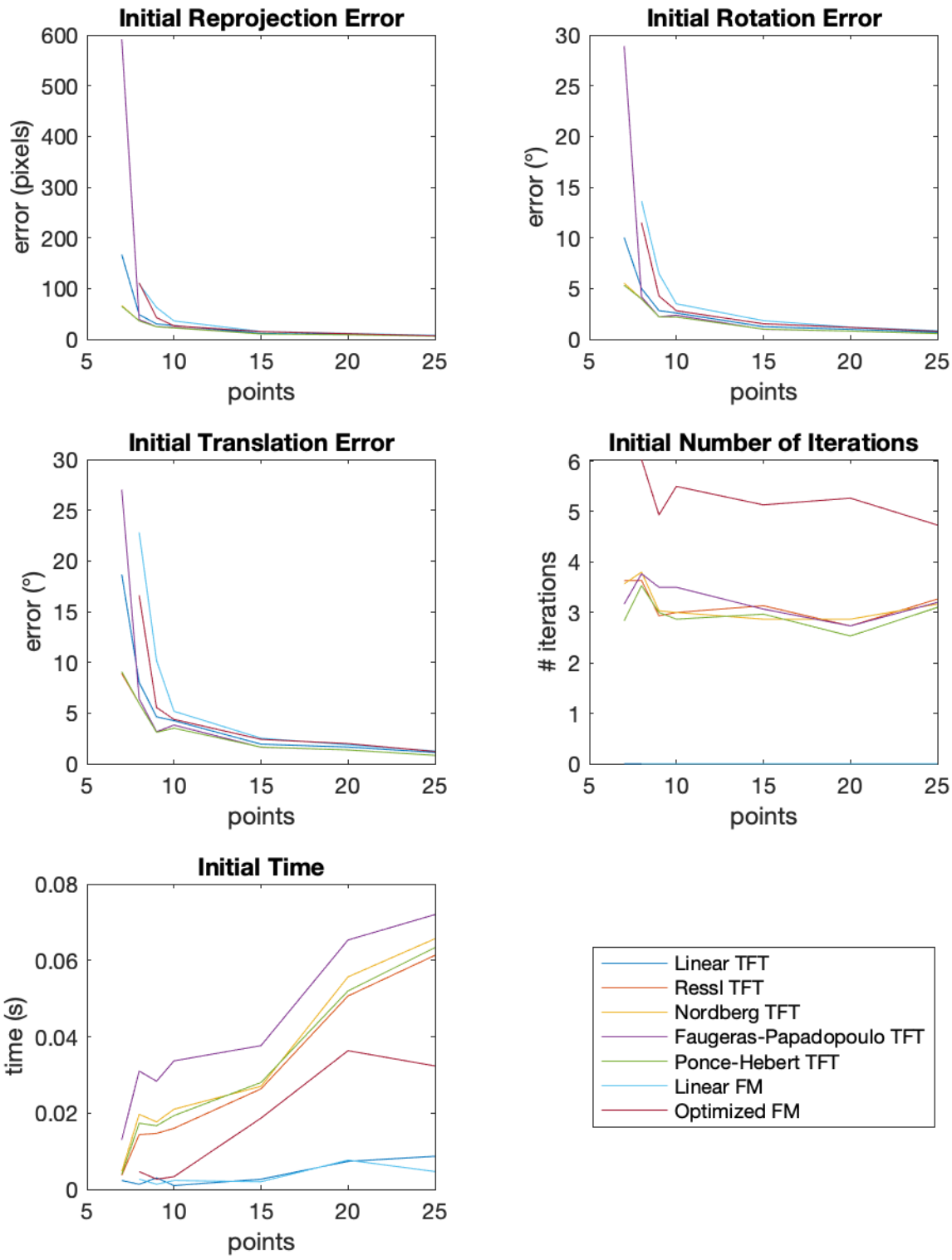


Figure 7. Initial reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left); when varying the number of image points.

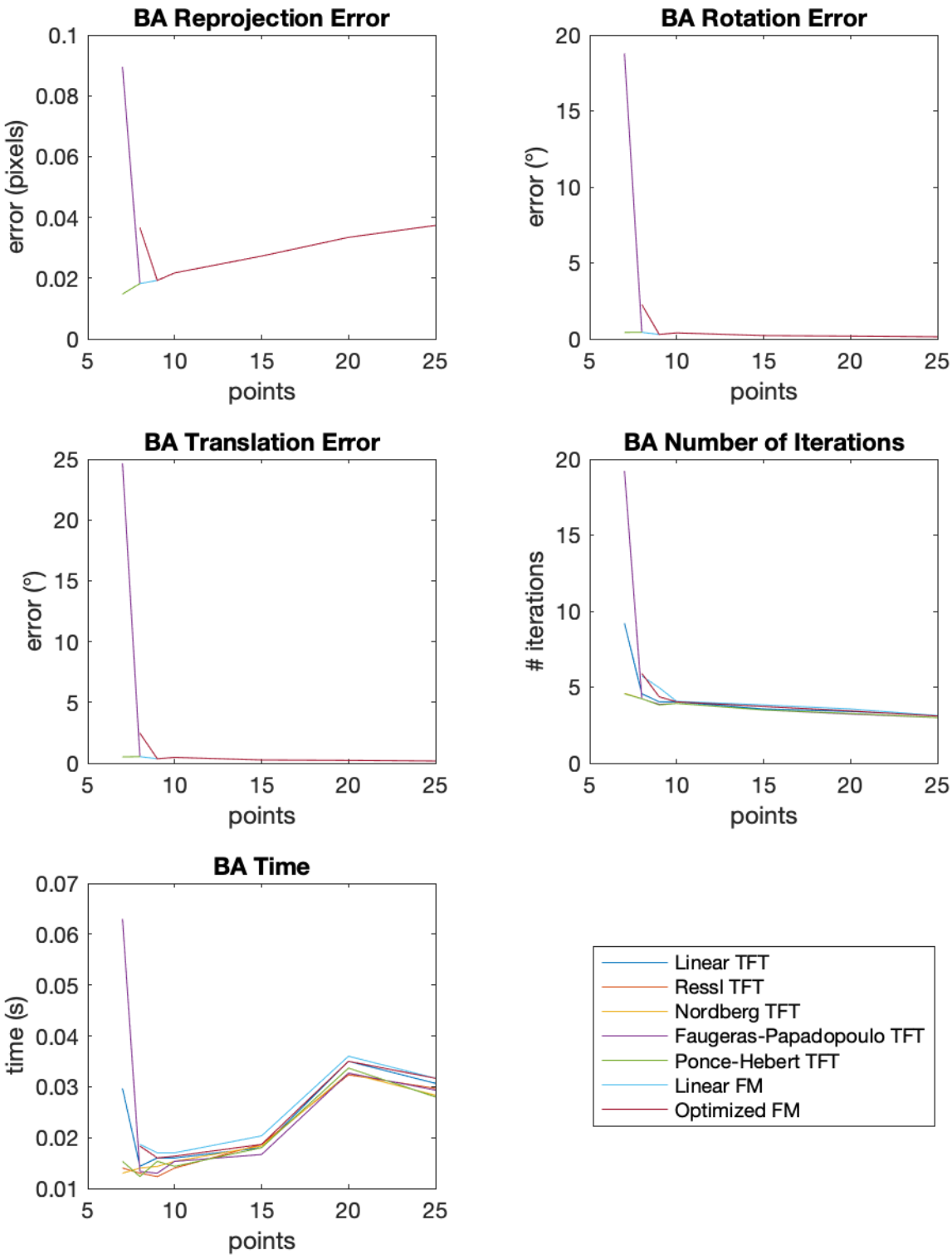


Figure 8. Reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left) after BA; when varying the number of image points.

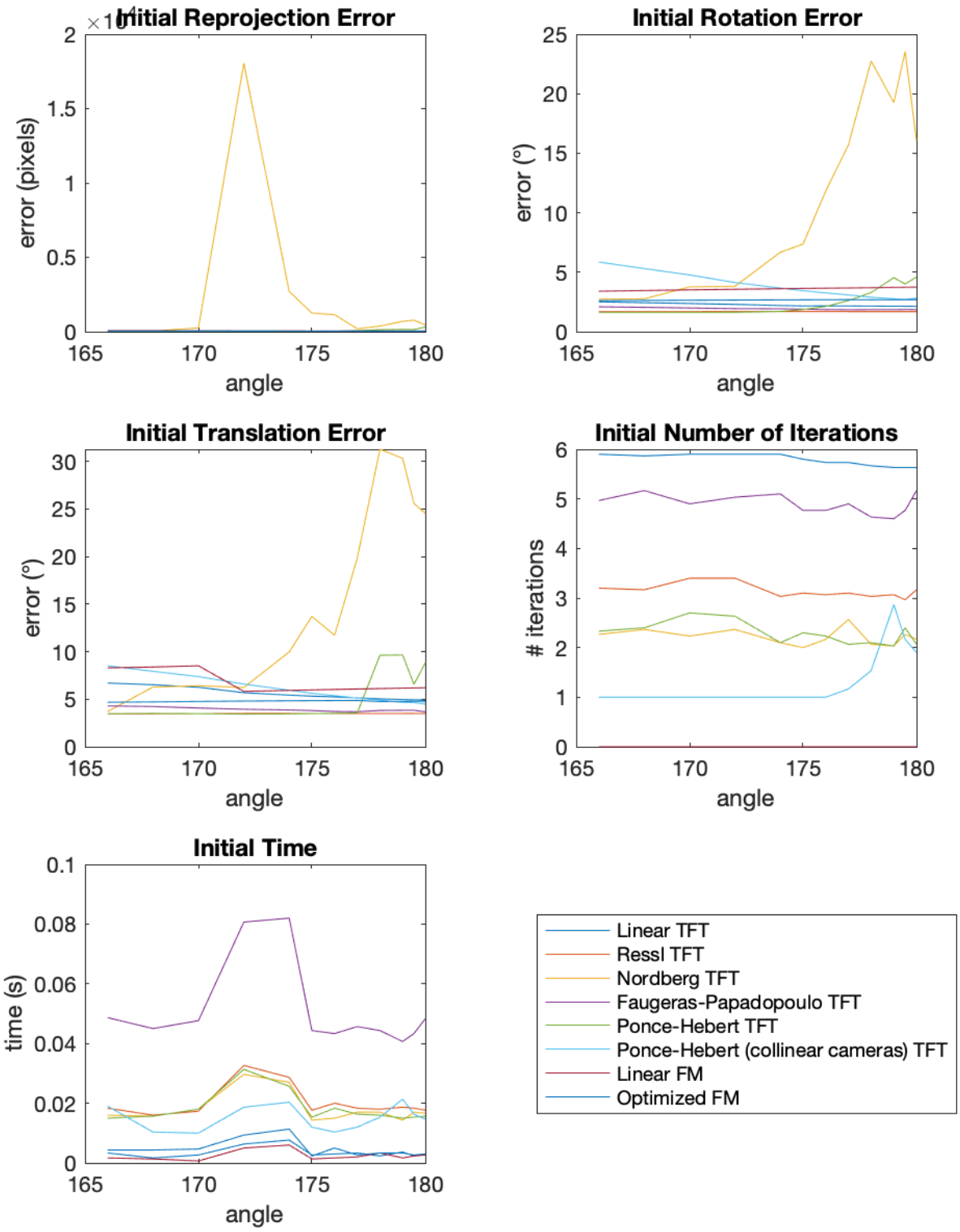


Figure 9. Initial reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left); when varying the angle among the three camera centers.

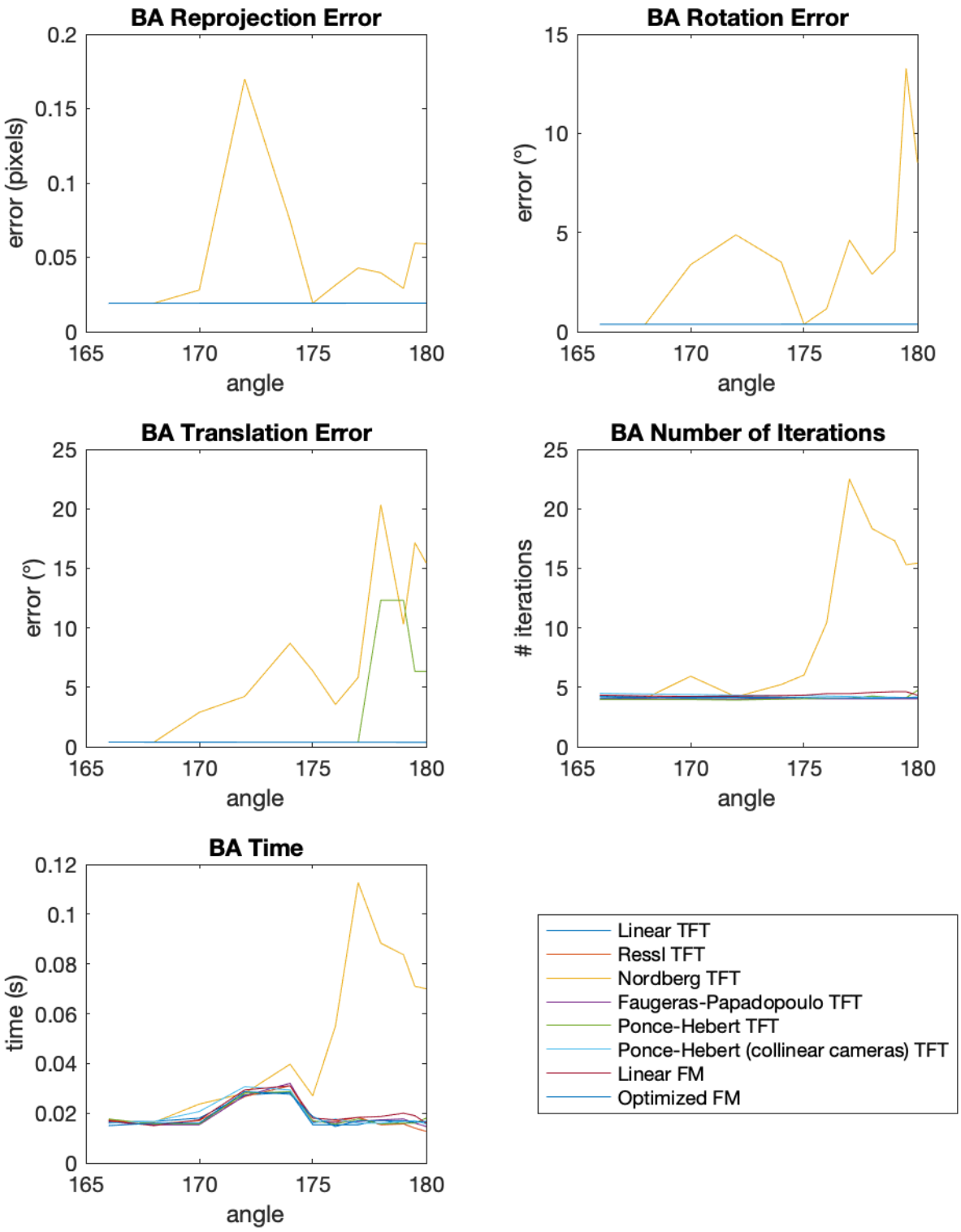


Figure 10. Reprojection error (top-left), rotation error (top-right), translation error (mid-left), number of iterations (mid-right), computational time (bottom-left) after BA; when varying the angle among the three camera centers.

5.2. Real Data

In assessing the efficacy of these methods within real-world contexts, we opted to utilize scenes from the EPFL Dense Multi-View Stereo Dataset, presented in [12], provided by the CVLab at EPFL. [3] [4]

Table (1) and (2) show metrics before and after BA with respect to the *fountain-P11* set of images from the dataset.

(a) (b) (c)

Figure 11. Generic three-view triplet of images with respect to *fountain-P11*. [12]

Table 1. Initial metrics with respect to the *fountain-P11* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	2.3953	0.1249	0.4048	0	0.0621
R-TFT	2.0474	0.1158	0.4003	2.8429	0.6400
N-TFT	2.1322	0.1334	0.4028	2.8000	0.6280
FP-TFT	2.3688	0.1187	0.4055	2.7714	0.6073
PH-TFT	2.0871	0.1167	0.4030	2.5857	0.5554
L-FM	1.9671	0.1149	0.3717	0	0.0273
O-FM	1.9530	0.1127	0.3658	4.9286	0.3209

Table 2. Metrics after BA with respect to the *fountain-P11* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	0.2814	0.0640	0.0743	3.8143	0.0743
R-TFT	0.2814	0.0640	0.0743	3.8286	0.0720
N-TFT	0.2814	0.0640	0.0743	3.8571	0.0716
FP-TFT	0.2814	0.0640	0.0743	3.8429	0.0723
PH-TFT	0.2814	0.0640	0.0743	3.8429	0.0743
L-FM	0.2814	0.0640	0.0743	3.7714	0.0816
O-FM	0.2814	0.0640	0.0743	3.8000	0.0784

[3] The EPFL Dense Multi-View Stereo Dataset, featuring the scenes utilized in our study, is readily accessible at the following location: https://documents.epfl.ch/groups/c/cv/cvlab-unit/www/data/multiview/denseMVS.html.

[4] Real trials are developed in the MATLAB script *RealExperiments.m* of the GitHub repository.

Table (3) and (4) show metrics before and after BA with respect to the *Herz-Jesu-P8* set of images from the dataset.

(a) (b) (c)

Figure 12. Generic three-view triplet of images with respect to *Herz-Jesu-P8*. [12]

Table 3. Initial metrics with respect to the *Herz-Jesu-P8* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	4.8062	0.4589	0.8707	0	0.0506
R-TFT	3.4792	0.3966	0.6677	2.7800	0.4904
N-TFT	4.0656	0.5252	0.6917	2.6600	0.4816
FP-TFT	4.5006	0.4459	0.8324	3.4400	0.5452
PH-TFT	4.5293	0.4261	0.6682	2.3000	0.4116
L-FM	3.7624	0.4142	0.7725	0	0.0224
O-FM	3.6503	0.4196	0.7654	5.6600	0.2906

Table 4. Metrics after BA with respect to the *Herz-Jesu-P8* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	0.3719	0.0635	0.0682	4.0600	0.0792
R-TFT	0.3719	0.0635	0.0682	4.0000	0.0674
N-TFT	0.3719	0.0635	0.0682	4.0400	0.0690
FP-TFT	0.3719	0.0635	0.0682	4.0600	0.0680
PH-TFT	0.3719	0.0635	0.0682	4.0000	0.0664
L-FM	0.3719	0.0635	0.0682	4.0000	0.0718
O-FM	0.3719	0.0635	0.0682	4.0200	0.0724

Table (5) and (6) show metrics before and after BA with respect to the *entry-P10* set of images from the dataset.

(a) (b) (c)

Figure 13. Generic three-view triplet of images with respect to *entry-P10*. [12]

Table 5. Initial metrics with respect to the *entry-P10* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	3.8006	0.4144	0.8582	0	0.0572
R-TFT	3.1021	0.3872	0.6520	2.8200	0.5023
N-TFT	3.4590	0.4812	0.6833	2.7800	0.5144
FP-TFT	3.4480	0.4310	0.8246	3.5200	0.6340
PH-TFT	3.5078	0.4129	0.6619	2.4000	0.4389
L-FM	3.6451	0.4075	0.7103	0	0.0388
O-FM	3.6054	0.4018	0.7578	5.6800	0.3476

Table 6. Metrics after BA with respect to the *entry-P10* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	0.3479	0.0581	0.0634	4.0153	0.0811
R-TFT	0.3479	0.0581	0.0634	4.0242	0.0682
N-TFT	0.3479	0.0581	0.0634	4.0167	0.0708
FP-TFT	0.3479	0.0581	0.0634	4.0271	0.0714
PH-TFT	0.3479	0.0581	0.0634	4.0015	0.0679
L-FM	0.3479	0.0581	0.0634	3.9920	0.0734
O-FM	0.3479	0.0581	0.0634	4.0030	0.0751

Table (7) and (8) show metrics before and after BA with respect to the *castle-P19* set of images from the dataset.

(a) (b) (c)

Figure 14. Generic three-view triplet of images with respect to *castle-P19*. [12]

Table 7. Initial metrics with respect to the *castle-P19* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	3.7411	0.4235	0.8439	0	0.0614
R-TFT	2.8651	0.4147	0.6328	2.8400	0.5132
N-TFT	2.9337	0.4612	0.6740	2.7200	0.4922
FP-TFT	3.0164	0.4001	0.8125	3.5200	0.5478
PH-TFT	3.1450	0.4138	0.6574	2.4800	0.4227
L-FM	3.4568	0.3966	0.6913	0	0.0432
O-FM	3.5679	0.3810	0.7155	5.7800	0.4490

Table 8. Metrics after BA with respect to the *castle-P19* set of images.

	repr. error (px)	R error (°)	t error (°)	# iter.	time (s)
L-TFT	0.3583	0.0601	0.0657	4.0177	0.0827
R-TFT	0.3583	0.0601	0.0657	4.0211	0.0698
N-TFT	0.3583	0.0601	0.0657	4.0235	0.0716
FP-TFT	0.3583	0.0601	0.0657	4.0301	0.0721
PH-TFT	0.3583	0.0601	0.0657	4.0021	0.0684
L-FM	0.3583	0.0601	0.0657	3.9974	0.0733
O-FM	0.3583	0.0601	0.0657	4.0081	0.0742

Unsurprisingly, considering the results obtained with the synthetic experiments, the BA has a major influence on the performances with respect to the different mathematical structures considered (*e.g.*, FM, TFT, parametrizations of the TFT).

In fact, regardless of the pose estimation technique considered, all errors (post-BA) converge to a fixed value, which is significantly lower than all the corresponding error values computed with the different parametrizations.

6. Conclusions

In our study, we explored different methods for estimating the trifocal tensor and figuring out the positions of three separate views. After thorough testing, we found that even though the trifocal tensor is a valid method, it doesn't offer enough benefits over using fundamental matrices from pairs of views to make it the better choice.

Our results highlight the practical benefits of simplicity and efficiency. We recommend focusing on pairwise constraints using fundamental matrices and then refining the results with bundle adjustment techniques. It's important to mention that bundle adjustment consistently reduces errors significantly. In this approach, the first step is to establish pairwise constraints to determine the relative scales of translations, using image triplets mainly for this purpose.

6.1. Future Work

Another interesting direction for future analysis is to see if using the trifocal tensor gives better results when working with more than three views (*i.e.*, $n > 3$). In such multi-view stereo setups, the way image pairs and triplets are combined will likely have a big impact on the overall effectiveness.

Our work also showed something important about bundle adjustment optimisation. We discovered that even if we start the process from very different points, it can still end up at a good solution. This means there might be a bigger area where the optimisation works well, and we plan to look into this more in future studies.

References

[1] N. Canterakis. A minimal set of constraints for the trifocal tensor. In *Computer Vision - ECCV 2000*, pages 84–99, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. 8

[2] O. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of 3 views. pages 477 – 484, 02 1998. 8

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 6

[4] L. F. Julià and P. Monasse. A critical review of the trifocal tensor estimation. In M. Paul, C. H. Morimoto, and Q. Huang, editors, *Image and Video Technology - 8th Pacific-Rim Symposium, PSIVT 2017, Wuhan, China, November 20-24, 2017, Revised Selected Papers*, volume 10749 of *Lecture Notes in Computer Science*, pages 337–349. Springer, 2017. 8

[5] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 11

[6] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. 293: 133–135, 1981. 5

[7] F. Neitzel. Generalization of total least-squares on example of unweighted and weighted 2d similarity transformation. *Journal of geodesy*, 84:751–762, 2010. 4

[8] K. Nordberg. A minimal parameterization of the trifocal tensor. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1224–1230. IEEE, 2009. 8, 9

[9] T. Papadopoulo and O. Faugeras. A new characterization of the trifocal tensor. In *Computer VisionECCV'98: 5th European Conference on Computer Vision Freiburg, Germany, June, 2–6, 1998 Proceedings, Volume I 5*, pages 109–123. Springer, 1998. 8, 9

[10] J. Ponce and M. Hebert. Trinocular geometry revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2014. 8, 10

[11] C. Ressl. A minimal set of constraints and a minimal parameterization for the trifocal tensor. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):277–282, 2002. 8, 9

[12] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008. 22, 23, 24, 25

[13] P. H. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and vision Computing*, 15(8):591–605, 1997. 8