Radial Distortion Triangulation

Zuzana Kukelova
Visual Recognition Group, FEE, CTU in Prague
kukelova@cmp.felk.cvut.cz

Viktor Larsson
Department of Computer Science, ETH Zürich
viktor.larsson@inf.ethz.ch

Abstract

This paper presents the first optimal, maximal likelihood, solution to the triangulation problem for radially distorted cameras. The proposed solution to the two-view triangulation problem minimizes the ℓ_2-norm of the reprojection error in the distorted image space.

We cast the problem as the search for corrected distorted image points, and we use a Lagrange multiplier formulation to impose the epipolar constraint for undistorted points. For the one-parameter division model, this formulation leads to a system of five quartic polynomial equations in five unknowns, which can be exactly solved using the Gröbner basis method. While the proposed Gröbner basis solution is provably optimal; it is too slow for practical applications.

Therefore, we developed a fast iterative solver to this problem. Extensive empirical tests show that the iterative algorithm delivers the optimal solution virtually every time, thus making it an ℓ_2-optimal algorithm de facto. *It is iterative in nature, yet in practice, it converges in no more than five iterations. We thoroughly evaluate the proposed method on both synthetic and real-world data, and we show the benefits of performing the triangulation in the distorted space in the presence of radial distortion.*

1. Introduction

In the parlance of modern computer vision, to triangulate a point—given $n \geq 2$ camera projection matrices $\{\mathtt{P}_i\}_{i=1}^n$, $\mathtt{P}_i \in \mathbb{R}^{3\times4}$, and a set of image points $\{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{x}_i = [x_i, y_i, 1]^\top$—is to find $\mathbf{X} \in \mathbb{R}^4$, such that

$$\alpha_i \mathbf{x}_i = \mathtt{P}_i \mathbf{X},\ i = 1, \ldots, n, \quad \alpha_i \in \mathbb{R}, \tag{1}$$

i.e., such that the points $\mathbf{x}_i$ are the projections of the point $\mathbf{X}$ using the projection matrices $\mathtt{P}_i$ [14].

For a noise-less scenario, the triangulation problem becomes a trivial exercise in linear algebra. In the presence of noise, however, the n rays originating from the camera centers through the image points $\mathbf{x}_i$ do not generally intersect in the 3D space, *i.e.*, there is no 3D point $\mathbf{X}$ that would satisfy Eq. (1) for all $\mathbf{x}_i$. Thus, for noisy data, the triangulation problem becomes an optimization problem of finding a point $\mathbf{X}$ that fits the constraints in Eq. 1 "the best". What constitutes "the best" fit necessarily depends on the input data and computational resources at hand. However, it has been shown [13] that assuming independent Gaussian noise on the image measurements, the optimal, maximum likelihood solution to the triangulation problem is a solution that minimizes the ℓ_2-norm of the reprojection error.

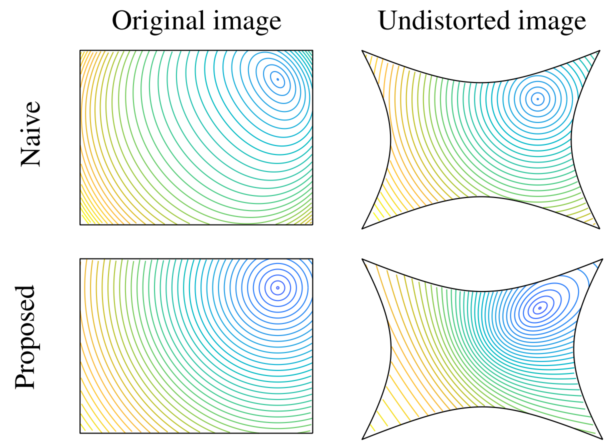


Figure 1. Level-sets for the reprojection error in the distorted and undistorted images. *Top:* First undistorting the images and then performing triangulation (*e.g.* using [25]) minimizes the ℓ_2 error in the undistorted images. Due to the non-linear distortion this does not minimize any meaningful cost in the original image. *Bottom:* The proposed method instead minimizes the ℓ_2 error in the original image.

Since triangulation is an integral part of many larger computer vision methods and systems, a plethora of algorithms for solving this problem has been proposed in the past. Taxonomy of the triangulation methods can be established along several lines: the methods may vary in the number of views they can handle, in the form of the objective function (algebraic, reprojection error), in the way they measure the error (ℓ_2-norm, ℓ_∞-norm, ℓ_1-norm), or in the optimization method they use to compute the results.

One of the simplest solutions to the triangulation problem is the linear least square method [14]. This method is fast and easily expandable to the multiview case, however, the method does not guarantee an optimal solution and it is prone to scaling issues. Usually, a solution provided by the linear least square method or by methods based on ℓ_∞-norm minimization [12, 19, 17] is used as initialization for a non-linear refinement method to be eventually optimized alongside other relevant parameters. Such an approach falls in the class of methods known as bundle adjustment [29, 2]. The bundle adjustment method is quite effective, yet it is still a local optimization method that requires good initial estimates. Inaccurate initialization may cause the method to get trapped in a local minimum.

A great deal of research effort went in recent years into developing globally optimal triangulation methods.

The first ℓ_2-norm optimal triangulation method is due to Hartley and Sturm [13]. This method is quite simple for two views, where the problem reduces to a solution to a polynomial equation of degree 6. However, it can't be easily extended to more views. In [18], Kanatani *et al*. introduced a fast iterative solution for the two-view triangulation problem. While their solution may fall into local minima, depending on the number of iterations, an extension of the method by Lindstrom [25] usually converges to the global optimum in two steps.

Optimal triangulation in three-views was solved for the first time by Stéwenius *et al*. [28]. This method, like its subsequent extensions and speedups [8, 9, 10], solves the problem using advanced algebraic methods for solving polynomial equations by searching for stationary points of an unconstrained cost function. Unfortunately, not only are these solutions too slow for any practical use, but also the implementation is quite involved. Kukelova *et al*. [22] presented a faster algorithm for three-view triangulation, however, their approach is only a relaxed formulation, thus not guaranteeing optimal solutions.

The last group of optimal triangulation methods is comprised of algorithms capable of handling an arbitrary number of views, *i.e*., multiview triangulation algorithms. These are usually based on branch-and-bound [12, 1, 26, 16], or second-order cone programming [17, 19, 3] approaches.

All previously mentioned solutions to the triangulation problem assume the pinhole camera model (1) without modeling radial distortion. However, nowadays consumer photography is dominated by mobile-phone and wide-angle action cameras (*e.g*. GoPro-type cameras). Therefore, images with significant radial lens distortion are quite common. In the presence of radial distortion, the projection equations (1) holds for undistorted image points

$$\alpha_i u(\mathbf{x}_i) = \mathtt{P}_i\mathbf{X},\ i = 1,\dots,n, \quad \alpha_i \in \mathbb{R}, \tag{2}$$

where $u(\cdot)$ is a non-linear undistortion function that undistorts the measured distorted image points.

Since the state-of-the-art triangulation methods cannot handle undistortion functions, a standard approach to the triangulation in the presence of radial distortion is to first undistort the image points and then run a triangulation method on the undistorted points. This means that the state-of-the-art "ℓ_2 optimal" triangulation methods minimize ℓ_2-norm of the reprojection error in the undistorted image space. However, assuming independent Gaussian noise on the original distorted image measurements, the optimal, maximal likelihood solution to the triangulation problem is a solution that minimizes ℓ_2-norm of the reprojection error in the distorted space. Therefore the state-of-the-art methods are not optimal in the presence of radial distortion.

In this paper, we propose the first solution to the two-view triangulation problem that is based on the minimization of the ℓ_2-norm of the reprojection error in the original image space, *i.e*. the distorted space. This method is the first optimal, maximal likelihood solution to the triangulation problem for radially distorted cameras.

We derived two solutions to this problem. The first solution, called GBD, is based on the Gröbner basis method for solving polynomial equations and it solves the problem by searching for all stationary points of an unconstrained cost function. Unfortunately, since the cost function results in a quite complicated system of polynomial equations, this solution is too slow for practical use.

Therefore we developed an iterative algorithm, called ITD through the rest of this paper. This algorithm significantly outperforms GBD in terms of speed. By extensive experimental comparison to the theoretically optimal GBD solver, we found ITD to deliver the optimal solution virtually every time, thus making it an ℓ_2-optimal algorithm *de facto*. It is iterative in nature, yet in practice, it converges in no more than five iterations.

Next, we formally introduce the triangulation problem for radially distorted image points.

2. Radial distortion triangulation

Let us formalize triangulation as a problem of reprojection error minimization in ℓ_2-norm in the distorted image space:

Problem 1 *Given the fundamental matrix* $\mathtt{F}$ *between the* 1-st *and the* 2-nd *view, and given two corresponding distorted image points* $\mathbf{x}_{d_i} = [x_{d_i}, y_{d_i}, 1]^\top$, $i = 1, 2,$

$$\begin{aligned} &\textit{minimize} \;\; f(\hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2}) = \textstyle\sum_{i=1}^{2} \|\mathbf{x}_{d_i} - \hat{\mathbf{x}}_{d_i}\|^2, \\ &\textit{subject to} \;\; u(\hat{\mathbf{x}}_{d_1})^\top \mathtt{F}\, u(\hat{\mathbf{x}}_{d_2}) = 0, \end{aligned}$$

where $\hat{\mathbf{x}}_{d_i} = [\hat{x}_{d_i}, \hat{y}_{d_i}, 1]^\top$ *are corrected distorted image points and* $u(\cdot)$ *is an undistortion function.*

Notice that instead of projection matrices $\mathtt{P}_i$ and 3D point $\mathbf{X}$ as in the case of the projection equation (2), Problem 1 formulates the triangulation constraint using the fundamental matrix $\mathtt{F}$, the corrected distorted image points $\hat{\mathbf{x}}_{d_i}$

and the undistortion function $u(\cdot)$. This formulation does not contain divisions needed for perspective projection and directly leads to polynomial constraints. The fundamental matrix $\mathtt{F}$ can be easily computed from $\mathtt{P}_i$ [14] and the triangulated point $\mathbf{X}$ from the corrected and thus noiseless image points $\hat{\mathbf{x}}_{d_i}$ and the undistortion function $u(\cdot)$.

Formally, Problem 1 is a problem of function minimization subject to equality constraints. Such a problem can be solved by transforming the original constrained optimization problem into an unconstrained problem by the method of Lagrange multipliers. In the case of Problem 1, this leads to the Lagrange function $L(\hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2}, \lambda)$:

$$L = \sum_{i=1}^{2} \|\mathbf{x}_{d_i} - \hat{\mathbf{x}}_{d_i}\|^2 + 2\lambda\, u(\hat{\mathbf{x}}_{d_1})^\top \mathtt{F}\, u(\hat{\mathbf{x}}_{d_2}), \quad (3)$$

where λ is the Lagrange multiplier and the constant '2' is introduced only for easier subsequent manipulation of the equations and it does not influence the final solution.

The theory of Lagrange multipliers tells us that if $f(\hat{\mathbf{x}}^\star_{d_1}, \hat{\mathbf{x}}^\star_{d_2})$ is a minimum of the original constrained Problem 1, then there exists $\lambda^\star$ such that $(\hat{\mathbf{x}}^\star_{d_1}, \hat{\mathbf{x}}^\star_{d_2}, \lambda^\star)$ is a stationary point of L (3), *i.e.*, a point where all the partial derivatives of L vanish. The Lagrange function $L(\hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2}, \lambda)$ in (3) is a function of five unknowns: four image point coordinates $\hat{\mathbf{x}}_{d_i}$, $i = 1, 2$, and the Lagrange multiplier λ. Thus, to find all stationary points of L we need to solve the following system of five polynomial equations in five unknowns:

$$\begin{aligned} u(\hat{\mathbf{x}}_{d_1})^\top \mathtt{F}\, u(\hat{\mathbf{x}}_{d_2}) &= 0, & (4)\\ 2\,\mathtt{S}(\hat{\mathbf{x}}_{d_1} - \mathbf{x}_{d_1}) + 2\lambda \mathtt{D}_{\hat{\mathbf{x}}_{d_1}} \mathtt{F}\, u(\hat{\mathbf{x}}_{d_2}) &= \mathbf{0}, & (5)\\ 2\,\mathtt{S}(\hat{\mathbf{x}}_{d_2} - \mathbf{x}_{d_2}) + 2\lambda \mathtt{D}_{\hat{\mathbf{x}}_{d_2}} \mathtt{F}^\top u(\hat{\mathbf{x}}_{d_1}) &= \mathbf{0}, & (6)\end{aligned}$$

where $\mathtt{S}$ is a $2{\times}3$ matrix that returns the first two coordinates of a three dimensional vector, and $\mathtt{D}_{\hat{\mathbf{x}}_{d_1}}$ and $\mathtt{D}_{\hat{\mathbf{x}}_{d_1}}$ are gradients of the undistortion functions $u(\hat{\mathbf{x}}_{d_1})$ and $u(\hat{\mathbf{x}}_{d_2})$.

Thanks to its compactness and expressive power, the one-parameter division model [11] is widely used to model radial lens distortion, and many different camera geometry solvers based on this model were proposed recently [7, 15, 6, 21, 20, 24, 27]. In the division model the undistortion function $u(\cdot)$ has the following form

$$u(\mathbf{x}_{d_i}) = \left[x_{d_i}, y_{d_i}, 1 + k(x^2_{d_i} + y^2_{d_i})\right]^\top, \quad (7)$$

where (x_{d_i}, y_{d_i}) are the centered distorted image coordinates and k is the distortion parameter.

For the one-parameter division model (7) the gradients $\mathtt{D}_{\hat{\mathbf{x}}_{d_1}}$ and $\mathtt{D}_{\hat{\mathbf{x}}_{d_2}}$ in (5) and (6) have the following form

$$\mathtt{D}_{\hat{\mathbf{x}}_{d_1}} = \begin{bmatrix} 1 & 0 & 2k_1\hat{x}_{d_1} \\ 0 & 1 & 2k_1\hat{y}_{d_1} \end{bmatrix}, \; \mathtt{D}_{\hat{\mathbf{x}}_{d_2}} = \begin{bmatrix} 1 & 0 & 2k_2\hat{x}_{d_2} \\ 0 & 1 & 2k_2\hat{y}_{d_2} \end{bmatrix}, \quad (8)$$

where k_1 and k_2 are the distortion parameters of the first and the second camera and $(\hat{x}_{d_i}, \hat{y}_{d_i})$ are the coordinates of corrected distorted image points. Note that by corrected points we do not mean undistorted points, but distorted points that after undistortion satisfy the epipolar constraint.

The solution for the two-parameter polynomial distortion model is described in the supplementary material.

2.1. Gröbner Basis solution

Equations (5) and (6) are vector equations obtained as partial derivatives of L w.r.t. the elements of $\hat{\mathbf{x}}_{d_i}$. Therefore (4)–(6) is a system of five quartic equations in five unknowns. This system can be solved by algebraic methods such as the Gröbner basis method. Using the automatic generator from Larsson et al. [23] we created a polynomial solver for the equations (4)–(6). In general, this system has 20 solutions and the generated solver performs linear elimination on a matrix of size 408×428. While this solver is guaranteed to return the globally optimal solution (up to numerical instabilities), it is too slow to be useful in practice. However, we will use it in Section 3.2 to validate the results of our iterative approach presented in the next section.

2.2. Iterative solution

In this section we propose an iterative solver that efficiently solves the original ℓ_2-optimal problem 1—the ITD solver.

First, let us denote $\Delta\mathbf{x}_{d_i} = \mathtt{S}\,(\mathbf{x}_{d_i} - \hat{\mathbf{x}}_{d_i})$ and

$$\begin{aligned} \mathbf{n}_1 &= \mathtt{D}_{\hat{\mathbf{x}}_{d_1}} \mathtt{F}\, u(\hat{\mathbf{x}}_{d_2}), & (9)\\ \mathbf{n}_2 &= \mathtt{D}_{\hat{\mathbf{x}}_{d_2}} \mathtt{F}^\top u(\hat{\mathbf{x}}_{d_1}), & (10)\end{aligned}$$

Now, we can rewrite equations (4)–(6) as

$$\begin{aligned} u(\mathbf{x}_{d_1} - \mathtt{S}^\top \Delta\mathbf{x}_{d_1})^\top \mathtt{F}\, u(\mathbf{x}_{d_2} - \mathtt{S}^\top \Delta\mathbf{x}_{d_2}) = 0, & \quad (11)\\ \lambda \mathtt{D}_{\hat{\mathbf{x}}_{d_1}} \mathtt{F}\, u(\hat{\mathbf{x}}_{d_2}) = \lambda \mathbf{n}_1 = \Delta\mathbf{x}_{d_1}, & \quad (12)\\ \lambda \mathtt{D}_{\hat{\mathbf{x}}_{d_2}} \mathtt{F}^\top u(\hat{\mathbf{x}}_{d_1}) = \lambda \mathbf{n}_2 = \Delta\mathbf{x}_{d_2}, & \quad (13)\end{aligned}$$

and the cost function of Problem 1 can be restated as

$$f(\hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2}) \;=\; \sum_{i=1}^{2} \Delta\mathbf{x}^\top_{d_i} \Delta\mathbf{x}_{d_i}. \quad (14)$$

The proposed iterative solution to equations (11)–(13) follows the idea of the two-view triangulation method from [25]. First, let $\langle \mathbf{x}^{k-1}_{d_1}, \mathbf{x}^{k-1}_{d_2} \rangle$ denote the current best estimate of the corrected distorted image points $\langle \hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2} \rangle$ after the $(k{-}1)$-*th* iteration. The measured points $\mathbf{x}^0_{d_i} \equiv \mathbf{x}_{d_i}$, $i = 1, 2$ are used as initialization. In the k-*th* iteration, to get the updated estimates $\langle \mathbf{x}^k_{d_1}, \mathbf{x}^k_{d_2} \rangle$ the algorithm starts by replacing the optimal points $\langle \hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2} \rangle$ on the left-hand side of equations (12)–(13) by the current best estimates $\langle \mathbf{x}^{k-1}_{d_1}, \mathbf{x}^{k-1}_{d_2} \rangle$. This results in the expressions for $\Delta\mathbf{x}_{d_1}$ and $\Delta\mathbf{x}_{d_2}$, which are in turn substituted into equation (11). The updated equation (11) is a univariate polynomial in λ^k:

$$u\left(\mathbf{x}_{d_1} - \mathtt{S}^\top(\lambda^k \mathbf{n}^k_1)\right)^\top \mathtt{F}\, u\left(\mathbf{x}_{d_2} - \mathtt{S}^\top(\lambda^k \mathbf{n}^k_2)\right) = 0, \quad (15)$$

where

Algorithm 1 ITD: Iterative radial distortion triangulation

Input: Fundamental matrix $\mathtt{F}$,
Image points $\mathbf{x}_{d_i} = [x_{d_i}, y_{d_i}, 1]^\top$, $i = 1, 2$,
Distortion parameters k_1, k_2
ϵ_1, ϵ_2, *maxiter*

Output: Corrected points $\hat{\mathbf{x}}_{d_i} = [\hat{x}_{d_i}, \hat{y}_{d_i}, 1]^\top$, $i = 1, 2$ that solve Problem 1

$\mathit{found} \leftarrow 0, \quad k \leftarrow 1$
$\mathbf{x}^0_{d_1} \leftarrow \mathbf{x}_{d_1}, \quad \mathbf{x}^0_{d_2} \leftarrow \mathbf{x}_{d_2}$
while (**not** *found*) **and** ($k \leq \mathit{maxiter}$) **do**
  $\mathbf{n}^k_1 \leftarrow \mathtt{D}_{\hat{\mathbf{x}}^{k-1}_{d_1}} \mathtt{F}\, u(\mathbf{x}^{k-1}_{d_2}), \quad \mathbf{n}^k_2 \leftarrow \mathtt{D}_{\hat{\mathbf{x}}^{k-1}_{d_2}} \mathtt{F}^\top u(\mathbf{x}^{k-1}_{d_1})$
  $\lambda^k \leftarrow$ Solution to equation (15), such that equation (14) is minimized
  $\Delta\mathbf{x}^k_{d_1} \leftarrow \lambda^k \mathbf{n}^k_1, \quad \Delta\mathbf{x}^k_{d_2} \leftarrow \lambda^k \mathbf{n}^k_2$
  $\mathit{err}^k \leftarrow \sum_{i=1}^2 \Delta\mathbf{x}^{k\top}_{d_i} \Delta\mathbf{x}^k_{d_i}$
  $\mathbf{x}^k_{d_1} \leftarrow \mathbf{x}_{d_1} - \Delta\mathbf{x}^k_{d_1}, \quad \mathbf{x}^k_{d_2} \leftarrow \mathbf{x}_{d_2} - \Delta\mathbf{x}^k_{d_2}$
  if ($k > 1$ **and** $\frac{|\mathit{err}^k - \mathit{err}^{k-1}|}{\mathit{err}^k} < \epsilon_1$) **or** ($\mathit{err}^k < \epsilon_2$) **then**
    $\langle \hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2} \rangle \leftarrow \langle \mathbf{x}^k_{d_1}, \mathbf{x}^k_{d_2} \rangle$
    $\mathit{found} \leftarrow 1$
  else
    $k \leftarrow k + 1$
  end if
end while
if not *found* **then**
  $\langle \hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2} \rangle \leftarrow \langle \mathbf{x}^{k-1}_{d_1}, \mathbf{x}^{k-1}_{d_2} \rangle$
end if

$$\mathbf{n}^k_1 = \mathtt{D}_{\hat{\mathbf{x}}^{k-1}_{d_1}} \mathtt{F}\, u(\mathbf{x}^{k-1}_{d_2}), \tag{16}$$

$$\mathbf{n}^k_2 = \mathtt{D}_{\hat{\mathbf{x}}^{k-1}_{d_2}} \mathtt{F}^\top u(\mathbf{x}^{k-1}_{d_1}). \tag{17}$$

For the one-parameter division model (7) the equation (15) is a 4^{th} degree polynomial in the unknown Lagrange multiplier λ^k. The roots of this polynomial can be easily found in closed form or using numerical methods. More details on the implementation of the ITD solver are in Section 2.3.

From the up to four possible solutions of (15), a solution that minimizes the cost function (14) is selected. Further, this solution is used to compute the k-*th* iteration displacements $\Delta\mathbf{x}^k_{d_1}$ and $\Delta\mathbf{x}^k_{d_2}$ using equations (12)–(13) and subsequently the new current estimates $\langle \mathbf{x}^k_{d_1}, \mathbf{x}^k_{d_2} \rangle$. The iteration stops based on two natural criteria: once the relative error of two consecutive iterations is smaller than some predetermined threshold $\epsilon_1 > 0$, or once the error itself is smaller than some predetermined threshold $\epsilon_2 > 0$.

Note that by solving the equation (15), we explicitly enforce the epipolar constraints in each iteration. ITD solver is formalized in Algorithm 1.

2.3. Implementation details and runtime

We have implemented the iterative ITD solver in C++. To solve the update in λ we use the closed form solution for univariate quartic polynomials. Similarly to [25], the cost function is always minimized by by the λ with smallest magnitude. To see this, note that when we substitute (12)-(13) into (14) the cost function reduces to

$$f(\hat{\mathbf{x}}_{d_1}, \hat{\mathbf{x}}_{d_2}) = \lambda^2 \left(\mathbf{n}_1^T \mathbf{n}_1 + \mathbf{n}_2^T \mathbf{n}_2\right). \tag{18}$$

With this in mind, we constructed another solver of (15) which instead of solving the quartic polynomial fully, performs Newton iterations starting from $\lambda = 0$. This typically convergences in less than five iterations and is slightly faster than solving the full quartic. Note that we are not performing local optimization on the cost function, but doing root refinement on the polynomial equation (15). In Section 3.2 we compare the two approaches and show that the Newton-based solver has similar performance as the quartic solver.

Since there was no implementation available from Lindstrom [25], we have reimplemented the method in C++. The runtimes for the different solvers are shown in Table 1. Note that our solver (and the solver from Lindstrom [25]) only return the corrected image points which satisfy the epipolar constraints. The runtime in Table 1 is only for the solver and not computing the 3D point.

	Our (Closed)	Our (Newton)	Lindstrom [25]
Runtime (ns)	1190	141	52
10^6 points/second	0.84	7.1	19.2

Table 1. The table shows the mean runtime in nanoseconds and the number of million points triangulated per second.

3. Evaluation on synthetic data

We have studied the performance of the proposed iterative algorithm (ITD) on synthetically generated ground-truth 3D scenes. These scenes were created by first generating 2000 random image points in the first camera $\mathtt{P}_1 = [\mathtt{I} \,|\, \mathbf{0}]$. The radial distortion of this camera was set to $k = -0.3$, the focal length to $f = 1300$ px (1750 px respectively) and the image size to 3000 px $\times$ 3000 px. These settings approximately correspond to the parameters of GoPro Hero4 camera with the wide (medium) field-of-view setting. The corresponding 3D points were created by backprojecting the points to random depths chosen uniformly from the interval $[2, 20]$. The 3D points were then projected to the second camera with random feasible orientation and position and with the same internal parameters as the first camera. Finally, Gaussian noise with standard deviation σ was added to the image points.

Note that here $k = -0.3$ corresponds to the radial distortion parameter that was applied to calibrated image points,

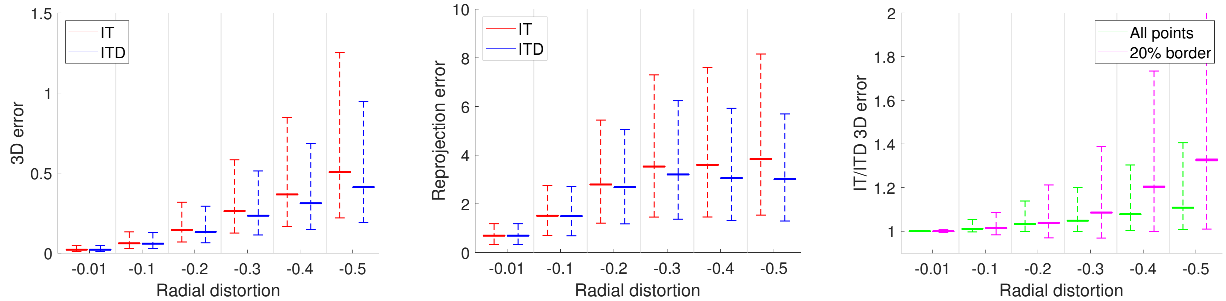


Figure 2. Comparison of the new ITD and the IT [25] solvers for varying radial distortions, 5% radial distortion error, 1 px image noise w.r.t. 3000 px $\times$ 3000 px image size, and $f = 1300$ px. The radial distortion $k = -0.3$ approximately corresponds to GoPro Wide setting.

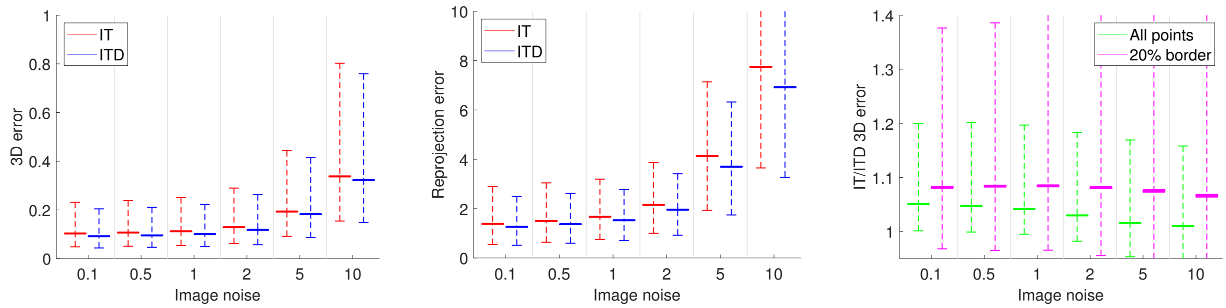


Figure 3. Comparison of the new ITD and the IT [25] solvers for varying image noise, $k = -0.3$, 2% radial distortion error, 3000 px $\times$ 3000 px image size and $f = 1300$ px. These camera parameters approximately correspond to the GoPro Wide setting.

which is a more common way of expressing radial distortion. For uncalibrated image points in (7), and assuming the calibration matrix to be $\mathtt{K} = diag([f, f, 1])$, this corresponds to the radial distortion parameter k/f^2.

In the first experiment, we tested the new triangulation solver on scenes with various noise contamination, different camera configurations, different radial distortions and different errors in the distortion parameter.

Figure 2 shows the result of our new iterative ITD solver and the state-of-the-art IT solver [25] for cameras with different radial distortions. In this case, we added $1px$ noise to image points and instead of the ground truth radial distortion parameter we used the distortion parameter with 5% error. This simulates a calibration error that can be present in real applications. Figure 2 shows the comparison of the 3D error, the reprojection error and the ratio of 3D errors of the IT [25] and the ITD solver on 1000 different scenes using box plots. For ratios of 3D errors, we also show the results for the 20% of points which have undergone the most distortion (*i.e.* points closest to the borders), to highlight the benefit of performing the triangulation in distorted space. It can be seen that especially for larger radial distortions our new method significantly outperforms the state-of-the-art IT solver [25], which minimizes ℓ_2 reprojection error in the undistorted image space. Note that here $k = -0.3$ approximately corresponds to GoPro Hero4 camera with the wide field-of-view setting.

A similar comparison for different image noise contamination is in Figure 3 and for radial distortion noise in Figure 4. In both these experiments, we set the radial distortion parameter to $k = -0.3$. It can be seen that in general the proposed method provides more accurate 3D point triangulations compared to the IT solver [25]. The improvement is even larger when we consider points closer to the image border which are more affected by the distortion.

More detailed statistics from these experiments including medians, means, the percentage of points where the new method gives smaller 3D error than [25], and the results for the "20% border points" are in the supplementary material.

3.1. Distance from the distortion center

In the second experiment, we studied the influence of the distance of the triangulated image points from the distortion center (image center) on the 3D triangulation error.

We generated 10 000 random scenes similarly to the previous experiment for the GoPro Wide setting ($f = 1300$ px) and the GoPro Medium setting ($f = 1750$ px) with 1 px image noise. In both cases the ground truth radial distortion w.r.t. to the calibrated image points was $k_{gt} = -0.3$, however, for the triangulation we used the distortion parameter $k = -0.29$, to simulate an error in the calibration.

Figure 5 shows medians and means of ratios of 3D errors of the state-of-the-art IT [25] and the new ITD method (*i.e.* 3D error IT / 3D error ITD) as a functions of distances d_1 and d_2 of the triangulated points $\mathbf{x}_{d_1}$ and $\mathbf{x}_{d_2}$ from the image center. Here w corresponds to the image width, which was in this case 3000 px. The top row shows results for GoPro Medium and the bottom row for GoPro Wide setting.

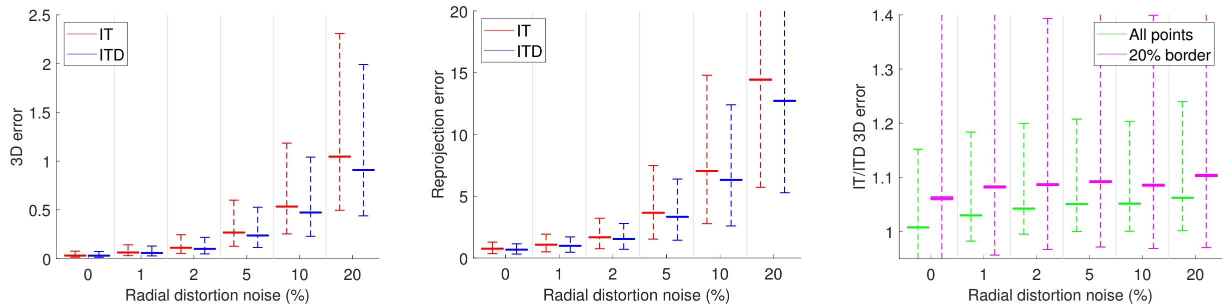
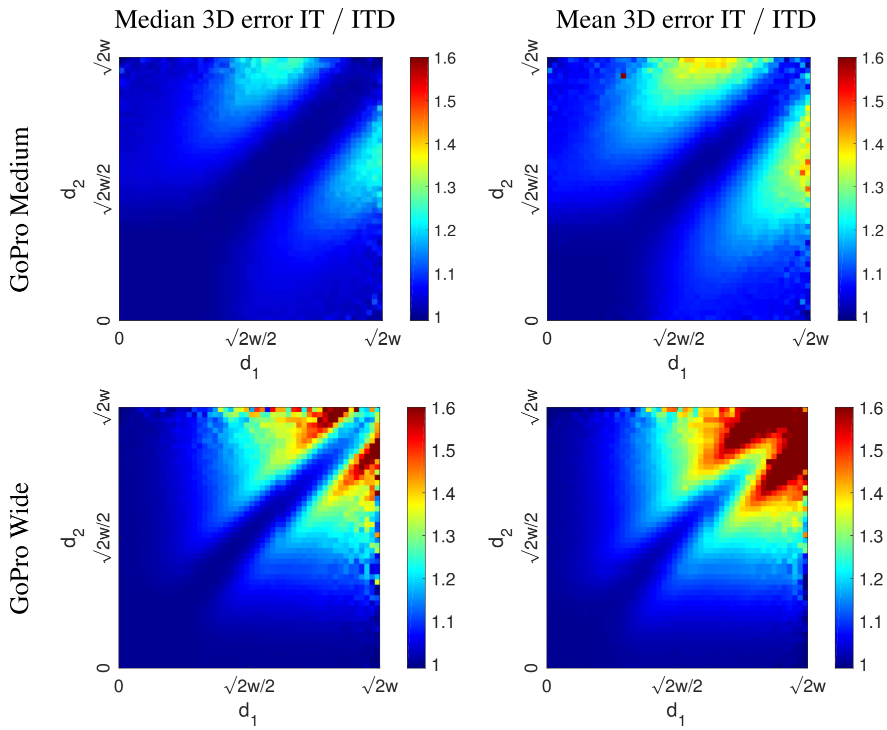


Figure 4. Comparison of the new ITD and the IT [25] solvers for varying radial distortion noise, $k = -0.3$, 1px image noise, 3000 px $\times$ 3000 px image size and $f = 1300$ px. These camera parameters approximately correspond to the GoPro Wide setting.



3D error

All points	mean	median
ITD	**0.1418**	**0.0703**
IT [25]	0.1529	0.0752
IT / ITD	**1.0789**	**1.0247**
ITD < IT	**71.4%**	

20% border	mean	median
ITD	**0.2695**	**0.1357**
IT [25]	0.2810	0.1476
IT / ITD	**1.1185**	**1.0392**
ITD < IT	**67.9%**	

All points	mean	median
ITD	**0.2946**	**0.1561**
IT [25]	0.3637	0.1739
IT / ITD	**1.1561**	**1.0503**
ITD < IT	**77.5%**	

20% border	mean	median
ITD	**0.6649**	**0.3855**
IT [25]	0.8723	0.4762
IT / ITD	**1.3465**	**1.1565**
ITD < IT	**79.3%**	

Figure 5. Medians and means of ratios of 3D errors of the IT [25] and the proposed ITD method (*i.e.* 3D error IT / 3D error ITD) as a functions of distances d_1 and d_2 of the triangulated points $\mathbf{x}_{d_1}$ and $\mathbf{x}_{d_2}$ from the image center for GoPro Medium and GoPro Wide settings.

The benefit of performing the triangulation in distorted space is visible especially for points that are further from the distortion center (*i.e.* points which have undergone larger distortion). Here for some points the 3D error of the state-of-the-art method [25], which optimizes ℓ_2 reprojection error in the undistorted space, was more than $1.6\times$ larger than the error of the new ITD method. Interestingly for points with similar distances from the distortion center, *i.e.* $d_1 \approx d_2$ (diagonals in graphs), the difference of the IT [25] and ITD method was not so significant.

More detailed statistics can be seen in the last column of Figure 5. Note that here we triangulated approximately $16M$ points, however, due to the way of generating our scene, not all combinations of distances (d_1, d_2) were equally present and therefore these graphs are not smooth.

3.2. Convergence of the iterative solver

In this section, we empirically show that the proposed iterative method essentially always converges to the globally optimal solution. To perform the experiment we used the optimal Gröbner basis solver from Section 2.1. To avoid incorrect solutions from numerical instabilities we performed further refinement on the solutions from the GBD solver.

For the experiment, we generated synthetic scenes similarly to Section 3. We compared the globally optimal solutions that we get from the GBD solver with the solutions found using the iterative approach ITD. The differences in the returned solutions are shown in Figure 6 (Left). Here we compared both approaches for solving quartic (15), *i.e.* the closed form solver for quartic and Newton iterations starting from $\lambda = 0$, as described in Section 2.3. We can see

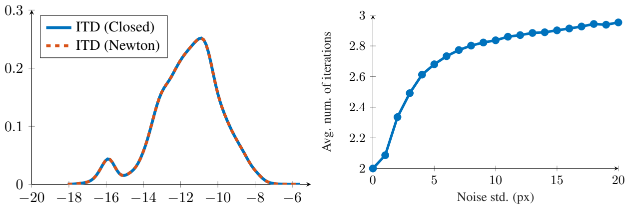


Figure 6. (Left) Log_{10} distance to global optima. (Right) The average number of iterations required for varying noise levels.

that both approaches are very stable and essentially always converge to the globally optimal solution. Figure 6 (Right) shows the number of iterations required for convergence for varying levels of image noise. Note that even for large noise levels (20 px) the proposed method typically converges in less than three iterations.

3.3. Comparison to bundle adjustment

In this section we compare with doing initial triangulation in the undistorted images followed by non-linear optimization of the reprojection error in the distorted images. Figure 7 shows a comparison of the runtime for synthetic data, using Levenberg-Marquardt for the non-linear refinement (maximum of 5 iterations). The experimental setup is similar to the one in Section 3. In the experiment we compare to linear triangulation (DLT [14]), the midpoint method (see [4, 13]) and the method from [25]. We also compare different methods for solving the linear least squares problem in the linear triangulation; SVD (for homogeneous parametrization) and QR/Cholesky (for inhomogeneous parametrization). For our method (ITD) and Lindstrom's method [25] (IT), the runtime includes solving the 3×3 linear system to recover the 3D points (which takes approximately 200 ns). We can see that our method clearly outperforms the competing methods. Note that LM needs to solve a 3×3 linear system in each iteration, making it significantly slower even in cases where it only runs two or three iterations. Figure 8 shows the percentage of instances where the methods failed to reach the optimal solution after local optimization (we consider it as a failure when $\|\mathbf{X} - \mathbf{X}^\star\|/\|\mathbf{X}^\star\| > 0.01$).

4. Evaluation on real images

In this section, we evaluate our approach on real images. We consider a set of images containing a checkerboard calibration pattern. This allows us to have reliable ground truth for 3D point positions. In the supplementary material we show additional experiments on images from general scenes.

4.1. Checkerboard calibration dataset

The dataset contains images of a checkerboard calibration pattern taken with a GoPro Hero4 camera. The camera was used in the medium (26 images) and wide (32 images) field-of-view settings (approximately 94 and 122 degrees horizontal field-of-view). The ground truth was created using the calibration toolbox from [5]. Since [5] estimates a polynomial distortion model, we refit the division model (7) using the estimated camera poses. The mean reprojection error for the calibration was 0.32 px and 0.41 px respectively. Some example images are shown in Figure 9.

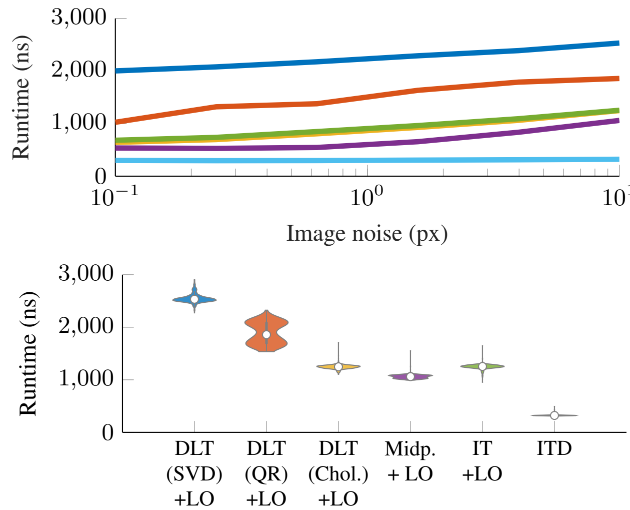


Figure 7. *Top:* Runtime vs. image noise. *Bottom:* Distributions of runtimes for $\sigma = 10$ px.

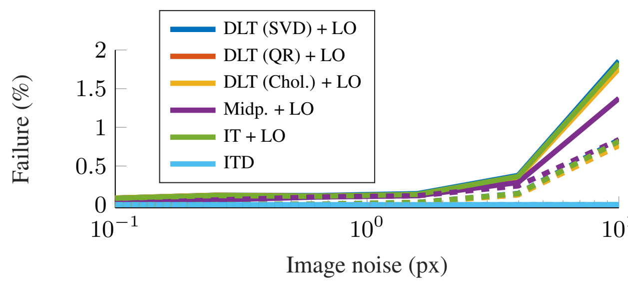


Figure 8. Failure to reach optimal solution after 5 iterations (dashed lines show 50 iterations).

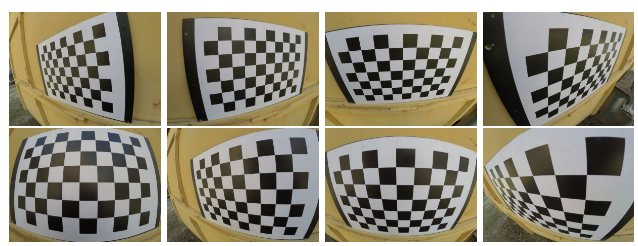


Figure 9. Example of checkerboard images used for the experiment in Section 4.1. *Top:* Medium field-of-view setting. *Bottom:* Wide field-of-view setting.

We again compared our ITD solver with the IT solver from [25] which performs triangulation on undistorted image points (*i.e.* is optimal in the undistorted space). For each pair of images, we computed the relative pose and performed the triangulation using both methods. The results are shown in Table 2 and 3. We also include results where

	3D error [mm]					Reprojection error [px]				
	mean		median			mean		median		
	IT	ITD	IT	ITD	ITD<IT	IT	ITD	IT	ITD	ITD<IT
0%	0.0794	**0.0793**	0.0637	**0.0635**	52.0%	0.1211	**0.1205**	0.0910	**0.0908**	100%
1%	0.0994	**0.0971**	0.0719	**0.0712**	60.5%	0.1709	**0.1699**	0.1217	**0.1213**	100%
5%	0.2728	**0.2563**	0.1472	**0.1416**	77.7%	0.5307	**0.5269**	0.2881	**0.2876**	100%
10%	0.5005	**0.4712**	0.2632	**0.2537**	82.8%	0.9865	**0.9788**	0.4953	**0.4940**	100%

Table 2. Results for the checkerboard experiment with the *medium* field-of-view setting.

	3D error [mm]					Reprojection error [px]				
	mean		median			mean		median		
	IT	ITD	IT	ITD	ITD<IT	IT	ITD	IT	ITD	ITD<IT
0%	0.1020	**0.1013**	0.0748	**0.0747**	51.6%	0.1902	**0.1853**	0.1439	**0.1407**	100%
1%	0.1774	**0.1603**	0.1078	**0.1026**	68.0%	0.3667	**0.3549**	0.2536	**0.2496**	100%
5%	0.7240	**0.6255**	0.3720	**0.3375**	83.7%	1.4505	**1.3970**	0.8300	**0.8151**	100%
10%	1.4032	**1.1967**	0.7254	**0.6512**	88.1%	2.8881	**2.7800**	1.6210	**1.5905**	100%

Table 3. Results for the checkerboard experiment with the *wide* field-of-view setting.

	3D error [mm]					Reprojection error [px]				
	mean		median			mean		median		
	IT	ITD	IT	ITD	ITD<IT	IT	ITD	IT	ITD	ITD<IT
0%	0.1339	**0.1319**	0.0993	**0.0966**	61.2%	0.1856	**0.1829**	0.1410	**0.1391**	100%
1%	0.2260	**0.2064**	0.1551	**0.1447**	74.8%	0.4158	**0.4097**	0.3294	**0.3275**	100%
5%	0.9282	**0.8086**	0.5185	**0.4732**	86.5%	1.7028	**1.6783**	1.3161	**1.2907**	100%
10%	1.6981	**1.5176**	1.1234	**1.0288**	88.0%	3.1144	**3.0666**	2.5120	**2.4858**	100%

Table 4. Results for the checkerboard experiment with the *medium* field-of-view setting. Top 5% of most distorted points.

	3D error [mm]					Reprojection error [px]				
	mean		median			mean		median		
	IT	ITD	IT	ITD	ITD<IT	IT	ITD	IT	ITD	ITD<IT
0%	0.2314	**0.2162**	0.1756	**0.1725**	61.8%	0.3273	**0.3064**	0.2441	**0.2324**	100%
1%	0.5222	**0.4180**	0.3272	**0.2752**	77.9%	0.8825	**0.8279**	0.7542	**0.7164**	100%
5%	2.2356	**1.7182**	1.3063	**1.0553**	88.1%	3.9506	**3.7037**	3.2601	**3.0760**	100%
10%	4.5794	**3.4405**	2.6298	**1.9991**	90.7%	8.1388	**7.6161**	6.3812	**6.0055**	100%

Table 5. Results for the checkerboard experiment with the *wide* field-of-view setting. Top 5% of most distorted points.

we simulated errors in the distortion parameter. In the table 5% error corresponds to the distortion parameter k being uniformly drawn from the interval $[0.95k_{GT}, 1.05k_{GT}]$.

To highlight the benefit of performing the triangulation in distorted space, we also show the results for the 5% of points which have undergone the most distortion (*i.e.* closest to the borders) separately in Table 4 and 5.

5. Conclusions

The paper presents the first optimal, maximal likelihood, solutions to the two-view triangulation problem for radially distorted cameras. The proposed solutions minimize the ℓ_2-norm of the reprojection error in the original distorted image space. The first proposed Gröbner basis solution, which searches for all stationary points of the Lagrange function is provably optimal, however, it is too slow for practical applications. The second iterative solver in practice converges in no more than five iterations to the optimal solution, thus making it an ℓ_2-optimal algorithm *de facto*.

We thoroughly evaluate the proposed method on both synthetic and real-world data, and we show the benefit of performing the triangulation in the distorted space in the presence of radial distortion.

Acknowledgments. Zuzana Kukelova was supported by the ESI Fund, OP RDE programme under the project International Mobility of Researchers MSCA-IF at CTU No. CZ.02.2.69/0.0/0.0/17_050/0008025. Viktor Larsson received funding from the ETH Zurich Postdoctoral Fellowship program and the Marie Sklodowska-Curie Actions COFUND program.

References

[1] Sameer Agarwal, Manmohan Chandraker, Fredrik Kahl, David Kriegman, and Serge Belongie. Practical global optimization for multiview geometry. In *European Conference on Computer Vision (ECCV)*, volume 1, pages 592–605, 2006. 2

[2] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. In *European Conference on Computer Vision (ECCV)*, pages 29–42. Springer, 2010. 2

[3] Chris Aholt, Sameer Agarwal, and Rekha R. Thomas. A QCQP approach to triangulation. In *European Conference on Computer Vision (ECCV)*, pages 654–667, 2012. 2

[4] Paul A Beardsley, Andrew Zisserman, and David W Murray. Navigation using affine structure from motion. In *European Conference on Computer Vision (ECCV)*, 1994. 7

[5] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`, 2004. 7

[6] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *Asian Conference on Computer Vision (ACCV)*, pages 11–24. Springer, 2010. 3

[7] Martin Byröd, Matthew A Brown, and Kalle Åström. Minimal solutions for panoramic stitching with radial distortion. In *British Machine Vision Conference (BMVC)*, pages 1–11, 2009. 3

[8] Martin Byröd, Klas Josephson, and Kalle Åström. Fast optimal three view triangulation. In *Asian Conference on Computer Vision (ACCV)*, 2007. 2

[9] Martin Byröd, Klas Josephson, and Kalle Åström. Improving numerical accuracy of gröbner basis polynomial equation solvers. In *International Conference on Computer Vision (ICCV)*, 2007. 2

[10] Martin Byröd, Klas Josephson, and Kalle Åström. A column-pivoting based strategy for monomial ordering in numerical gröbner basis calculations. In *European Conference on Computer Vision (ECCV)*, 2008. 2

[11] Andrew W Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2001. 3

[12] Richard Hartley and Frederik Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Computer Vision and Pattern Recognition (CVPR)*, 2004. 2

[13] Richard Hartley and Peter F. Sturm. Triangulation. *Computer Vision and Image Understanding (CVIU)*, 68(2):146–157, 1997. 1, 2, 7

[14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 1, 3, 7

[15] Klas Josephson and Martin Byrod. Pose estimation with radial distortion and unknown focal length. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2419–2426. IEEE, 2009. 3

[16] Klas Josephson and Fredrik Kahl. Triangulation of points, lines and conics. *Journal of Mathematical Imaging and Vision (JMIV)*, 32(2):215–225, 2008. 2

[17] Fredrik Kahl and Richard Hartley. Multiple-view geometry under the L_∞-norm. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 30(9):1603–1617, Sept 2008. 2

[18] Kenichi Kanatani, Yasuyuki Sugaya, and Hirotaka Niitsuma. Triangulation from two views revisited: Hartley-sturm vs. optimal correction. In *British Machine Vision Conference (BMVC)*, 2008. 2

[19] Qifa Ke and Takeo Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 29(10):1834–1847, 2007. 2

[20] Yubin Kuang, Jan E Solem, Fredrik Kahl, and Kalle Astrom. Minimal solvers for relative pose with a single unknown radial distortion. In *Computer Vision and Pattern Recognition (CVPR)*, pages 33–40, 2014. 3

[21] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *International Conference on Computer Vision (ICCV)*, pages 2816–2823, 2013. 3

[22] Zuzana Kukelova, Tomas Pajdla, and Martin Bujnak. Fast and stable algebraic solution to L_2 three-view triangulation. In *International Conference on 3D Vision (3DV)*, pages 326–333. IEEE, 2013. 2

[23] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[24] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *International Conference on Computer Vision (ICCV)*, pages 2335–2343. IEEE, 2017. 3

[25] Peter Lindstrom. Triangulation made easy. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2, 3, 4, 5, 6, 7

[26] Fangfang Lu and Richard Hartley. A fast optimal algorithm for L_2 triangulation. In *Asian Conference on Computer Vision (ACCV)*, 2007. 2

[27] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondrej Chum. Radially-distorted conjugate translations. In *Conference on Computer Vision and Pattern Recognition*, volume 2, 2018. 3

[28] Henrik Stéwenius, Frederik Schaffalitzky, and David Nistér. How hard is 3-view triangulation really? In *International Conference on Computer Vision (ICCV)*, volume 1, pages 686–693, 2005. 2

[29] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000. 2