Absolute and Relative Pose Estimation in Refractive Multi View

Xiao Hu[1,2], François Lauze[2], Kim Steenstrup Pedersen[2,3], and Jean Mélou[4]
[1]: DTU Space, Technical University of Denmark, Lyngby, Denmark
[2]: Department of Computer Science (DIKU), University of Copenhagen, Denmark
[3]: Natural History Museum of Denmark (NHMD), University of Copenhagen, Denmark
[4]: IRIT, UMR CNRS 5505, Toulouse, France
xiahaa@space.dtu.dk, francois@diku.dk, kimstp@di.ku.dk, jean.melou@toulouse-inp.fr

Abstract

This paper investigates absolute and relative pose estimation under refraction, which are essential problems for refractive structure from motion. We first present an absolute pose estimation algorithm by leveraging an efficient iterative refinement. Then, we derive a novel refractive epipolar constraint for relative pose estimation. The epipolar constraint is established based on the virtual camera transformation, making it in a succinct form and can be efficiently optimized. Evaluations of the proposed algorithms on synthetic data show superior accuracy and computational efficiency to state-of-the-art methods. For further validation, we demonstrate the performance on real data and show the application in 3D reconstruction of objects under refraction.

1. Introduction

Recovering camera pose is one of the major elements of Structure from Motion (SfM) and Multiview Stereo (MVS). When different transparent media are present in the light path, refraction occurs which bends the light trajectories thereby rendering classical methods incorrect [47]. Examples include imaging through a water surface, underwater imaging, or imaging objects encased in a transparent medium, as illustrated in Fig. 1. Refraction invalidates camera models [47] and SfM needs to be adapted to it. In the case of **planar interfaces**, which is the one we study in this work, several authors have proposed Refractive Structure from Motion (RSfM) approaches [21, 29, 13, 20, 4, 5] with adaptations to 3D reconstruction and endoscopy [18, 4, 5]. Several SfM functional modules need to be adapted to refraction: 1) geometric verification of feature matching; 2) absolute pose estimation; 3) relative pose estimation; 4) triangulation; 5) bundle adjustment. Two classical geometric objects, the essential matrix and the homography matrix do not hold under refraction [7]. They can however be used at the expense of removing some correct matches. Bundle adjustment requires solving a very large amount of expensive quartic reprojection equations [20]. Thus feature matching verification, triangulation and bundle adjustment are to some extent solved. But absolute and relative pose estimations remain challenging. The imaging system under refraction becomes an axial camera [3], a special case of the generalized camera (GC) model [12, 44]. Previously proposed methods for the GC model have been shown to be quite sensitive to noise, and therefore the camera pose cannot be accurately estimated even in low-noise conditions [5, 17].

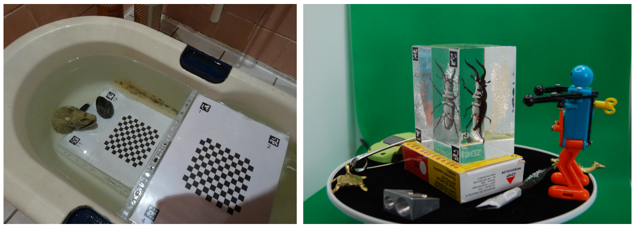


Figure 1. (Left) Bathtub scene - 1 out 4 images used in the camera pose estimation in Fig. 11. (Right) A stag beetle encased in crystal resin - 1 of 30 images used for SfM reconstruction in Fig. 10.

In this paper, we focus on absolute and relative pose estimation of a camera under refraction. Two scenarios can be distinguished. They are shown in Fig. 2; in **scenario 1** the refractive interface is fixed in the camera coordinate frame, and in **scenario 2** the refractive interface is fixed in the world coordinate frame. e.g. an object embedded in a transparent medium (resin, amber *etc.*) or underwater objects imaged above the surface. Most of the existing works [4, 5, 18, 20] target scenario 1, while [15] and our work focus on scenario 2. However, [15] targets only a very restricted subset of scenario 2 as the parameters of the refractive interface in each camera view are supposed to all be the same, equal to the parameters of the refractive in-

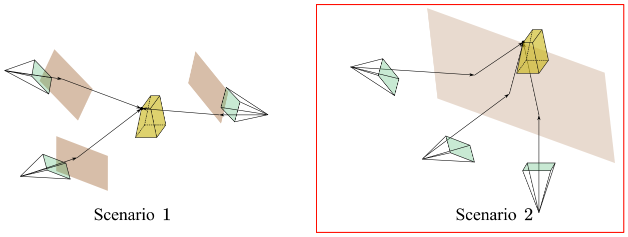


Figure 2. The considered scenarios: Scenario 1, the refractive interface is fixed in the camera coordinate frame and it moves together with the camera in the world coordinate frame, while in scenario 2, the refractive interface is fixed in the world coordinate frame.

terface of the reference view. By contrast, our work does not rely on such an assumption. We propose two iterative minimisation algorithms for absolute and relative pose estimation under refraction. Their formulation allows for easy derivations of the corresponding Jacobian matrices, which boosts computational efficiency. For relative pose estimation, we first derive a compact epipolar constraint under refraction using a virtual camera transformation. It is embedded in an iterative refinement to relative pose. Experiments on synthetic and real data are used to evaluate the performance of the proposed algorithms. We compare our methods with state-of-the-art methods for noise robustness, accuracy, and computational efficiency. Experiments show the proposed method outperforms state-of-the-art methods. Finally, we use them to reconstruct real data objects imaged under refraction. In detail, our contributions are the following: **1)** An efficient, accurate and robust to noise absolute camera pose estimation method. **2)** A simpler (than [5]) formulation of the epipolar constraint under refraction. To the best of our knowledge, this is the first relative pose estimation algorithm proposed for scenario 2. **3)** Detailed experiments demonstrate the robustness, accuracy and performance of our method.

2. Related work

Absolute pose estimation for a single perspective camera is well studied with a number of solutions that achieve impressive accuracy, *e.g.* [26, 10, 31, 14, 8]. In the case of refraction, the geometry of a perspective camera no longer holds. A few works have tackled this problem; an eight-point algorithm to calibrate the refractive interface and estimate the camera pose was introduced in [3]. Using accelerometers to estimate the camera's vertical direction, a two-point algorithm to estimate translation heading angle was proposed [6], but limited to a horizontal refractive plane. A five-point algorithm using co-planarity constraints [3] is presented in [13]. An alternative pose and depth optimisation is proposed in [20]. A few papers investigate the absolute pose estimation problem for generalized cameras or multi-camera systems; [39, 23, 30, 35, 49] separately proposed a minimal solver for recovering the pose of a multi-camera system. Besides minimal solvers, non-minimal solvers [25, 23, 45, 9] have been proposed by solving polynomials using Gröbner basis solvers [28]. However, they cannot be applied in scenario 2.

Chari and Sturm [7] studied the two-view refractive geometry and derived corresponding refractive fundamental (rF) and homography matrices. A seven-point linear solution for relative pose estimation with known vertical direction from an accelerometer was proposed in [6]. When camera orientation is known, [22] solves the relative translation optimally under the L_∞-norm, and otherwise use an evolutionary algorithm for hybrid optimization. Unlike previous methods that require extra equipment or prior information, [20] introduces an iterative solution using the geometric constraints proposed by [3]. Their solution has been integrated into an RSfM for deep-sea 3D reconstruction [18], this relies on good initialisation. [4] shows that the two-view relationship under refraction can be established using the generalized epipolar constraint (GEC) [40]. Although the GEC problem has minimal solutions [43, 48] and non-minimal linear ones [32, 37, 24], they are particularly sensitive to noise [4, 5]. The same paper proposes a novel formulation of the rF constraint, of size 21 × 12. Experiments show that relative pose estimation based on the rF constraint outperforms other state-of-the-art techniques. However, feature-dependency and high dimensionality make their method time-consuming.

3. Notations and Preliminaries

We denote scalars by lowercase letters, and vectors (resp. matrices) are denoted by bold lowercase letters (resp. bold uppercase letters). The identity matrix is denoted $\mathbf{I}$. SO(3) is the group of rotations. Its Lie algebra $\mathfrak{so}$ of skew-symmetric matrices is isomorphic to $\mathbb{R}^3$, we denote it by $\vee$ and its converse by $\wedge$, following [34]. Objects in virtual camera coordinates are denoted with a $\cdot_{\mathrm{V}}$ subscript and the i-coordinate of an object $\mathbf{p}$ is denoted by $\mathbf{p}(i)$, and the list of i, j and k coordinates by $\mathbf{p}_{(}i, j, k)$. Our formulations make systematic use of Plücker coordinates [44].

3.1. Background on Refraction

Snell's law, $\mu_i \sin\theta_i = \mu_j \sin\theta_j$, describes the relationship between the angles of incidence θ_i and refraction θ_j from the normal $\mathbf{n}$ and the refractive indices μ_i and μ_j of the media. Setting $\lambda = \mu_i/\mu_j$, Snell's law in vector form is (Fig. 3):

$$\mathbf{r} = \lambda\mathbf{i} + \mathbf{n}\sqrt{1 - \lambda^2\left[1 - (\mathbf{n}\cdot\mathbf{i})^2\right]} - \lambda(\mathbf{n}\cdot\mathbf{i})\mathbf{n}\,, \qquad (1)$$

where $\mathbf{i}, \mathbf{r}$ are the direction of the incident and transmitted ray, respectively. Furthermore, along with the origin $\mathbf{t}$ of the

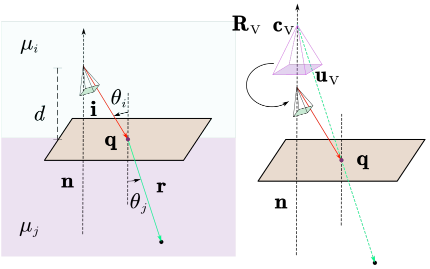


Figure 3. Illustration of Snell's law in vector form (left) and the idea of virtual camera coordinate (Right). The virtual camera satisfies the classical perspective projection model.

incident ray and the last interface parameter d (any point $\mathbf{p}$ on the plane satisfies $\mathbf{n}^\top \mathbf{p} + \text{d} = 0$), the point of refraction $\mathbf{q}$ can be computed as

$$\mathbf{q} = \left[\begin{array}{cc} (\mathbf{t} + \frac{|\mathbf{n}^\top \mathbf{t} + \text{d}|}{\bar{\mathbf{i}} \cdot \mathbf{n}} \bar{\mathbf{i}})^\top & 1 \end{array} \right]^\top , \tag{2}$$

where $\bar{\mathbf{i}}$ is the normalized vector of $\mathbf{i}$. These relations can be cascaded to flat multi-layer refraction, see [3]. We focus on the single-layer case though our proposed method can also be used for multiple refractions as long as the parameters of the refractive interfaces are known.

3.2. Virtual Cameras Under Refraction

One can find a virtual camera frame such that the correspondence between each 3D point and its 2D feature is given by a perspective projection, see [46, 19] and Fig. 3. The virtual camera center $\mathbf{c}_\text{V}$ is the intersection point of the refraction ray $\mathbf{r}$ and the line along the interface normal $\mathbf{n}$ through the real camera center $\mathbf{c}$

$$\mathbf{c}_\text{V} = \alpha_2 \mathbf{n} + \mathbf{c}, \text{ where } \mathbf{q} + \alpha_1 \mathbf{r} = \alpha_2 \mathbf{n} + \mathbf{c} \, . \tag{3}$$

The rotation matrix $\mathbf{R}_\text{V}$ from the camera coordinate frame is found by aligning the interface's normal to the optical axis (i.e., $\mathbf{e}_3 = [0, 0, 1]^\top$):

$$\mathbf{R}_\text{V} = \exp(\hat{\mathbf{a}}), \; \mathbf{a} = \theta \mathbf{l}, \; \theta = \mathbf{e}_3^\top \cdot \mathbf{n}, \; \mathbf{l} = \frac{\mathbf{e}_3^\top \times \mathbf{n}}{\|\mathbf{e}_3^\top \times \mathbf{n}\|} \, . \tag{4}$$

$\mathbf{R}_\text{V}$ is feature-independent; it only relies on the normal of the interface and the optical axis. $\mathbf{c}_\text{V}$ is however *feature-dependent*, it also depends on $\mathbf{r}$. The virtual focal length f_V is the distance d between the camera center and the refractive plane [20]. The relation between a 3D point $\mathbf{p}_\text{V}$ in virtual camera coordinates and its projection $\mathbf{u}_\text{V}$ becomes

$$\mathbf{u}_\text{V} = f_\text{V} \left[\begin{array}{cc} \frac{\mathbf{p}_\text{V}(1)}{\mathbf{p}_\text{V}(3)} & \frac{\mathbf{p}_\text{V}(2)}{\mathbf{p}_\text{V}(3)} \end{array} \right]^\top \, . \tag{5}$$

4. Methods

In the rest of this work, we assume a single planar refractive interface. For comprehension, we first present the constraints used for absolute and relative pose refinement geometrically. Then, we detail the exact formulations in scenario 2 under the assumption that the refractive plane parameters in the world coordinate frame are known.

4.1. Contribution 1: Absolute Pose Refinement

The proposed absolute pose refinement is based on the geometric constraint shown in Fig. 4. Given a 3D point $\mathbf{p}$ and its corresponding image point $\mathbf{q}$, if the parameters of the refractive interface are known, the line $\mathbf{l}$ can be found. Clearly, $\mathbf{p}$ lies on $\mathbf{l}$, which serves as the geometric constraint for absolute pose refinement. We express it in terms of the Plücker coordinates for the line $\mathbf{l}$. Assuming the refractive point $\mathbf{q}$ and the refraction ray $\mathbf{r}$ are known, the Plücker coordinates of the line $\mathbf{l}$ parallel to the ray direction $\mathbf{r}$ and passing the refractive point $\mathbf{q}$ is given by

$$\mathbf{l} = \left[\begin{array}{c} \mathbf{q}(4)\mathbf{r}(1,2,3) - \mathbf{r}(4)\mathbf{q}(1,2,3) \\ \mathbf{q}(1,2,3) \times \mathbf{r}(1,2,3) \end{array} \right] . \tag{6}$$

$\mathbf{p}$ lies on $\mathbf{l}$ if and only if

$$\mathbf{l}(1,2,3) \times \mathbf{p} + \mathbf{l}(4,5,6) = \mathbf{0} \, . \tag{7}$$

It is note that, in scenario 2, $\mathbf{l}$ has a implicit dependency on the camera pose $(\mathbf{R}, \mathbf{c})$. In practice, (7) is enforced in a least-squares formulation which serves as an objective function for $(\mathbf{R}$,$\mathbf{c})$

$$\underset{\mathbf{R},\mathbf{c}}{\arg\min} \sum_{i=1}^{N} \|\mathbf{l}^i(1,2,3) \times \mathbf{p}^i + \mathbf{l}^i(4,5,6)\|^2 \, , \tag{8}$$

with i the index of an image point and N the total number of image points. Unlike [15] that minimizes the reprojection error or the distance between 3D points, the objective in (8) is a sum of square distances from points to a given line. This is a nonlinear optimization problem that can be solved iteratively.

4.2. Contribution 2: Relative Pose Refinement

In this paragraph, we introduce first an epipolar constraint which serves as basis for the relative pose refinement and then our optimization objective for this refinement.

4.2.1 Epipolar constraint from a new perspective

Previous works [7, 4, 5, 20, 18] all derive an epipolar constraint using the Klein quadric. Instead, we write the standard epipolar constraint, but using the virtual cameras. We use the co-planar constraint shown in Fig. 5. Given an arbitrary feature point in view 1 and its matched feature point

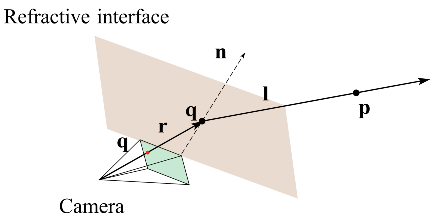


Figure 4. Geometric constraint used for absolute pose refinement. Given a 3D point $\mathbf{p}$ and its corresponding image point $\mathbf{q}$, if the parameters of the refractive interface are known, the line $\mathbf{l}$ can be found. The point $\mathbf{p}$ lies on $\mathbf{l}$, which is used as the geometric constraint for absolute pose refinement.

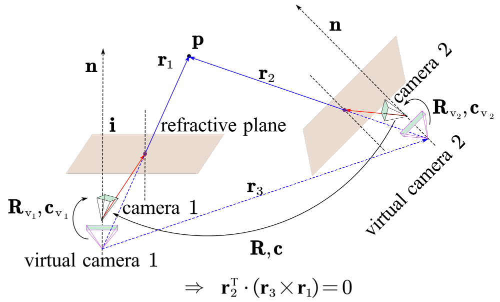

n
p
$\mathbf{r}_1$
$\mathbf{r}_2$
n
camera 2
$\mathbf{R}_{v_2}, \mathbf{c}_{v_2}$
i
refractive plane
virtual camera 2
$\mathbf{r}_3$
$\mathbf{R}_{v_1}, \mathbf{c}_{v_1}$
camera 1
$\mathbf{R}, \mathbf{c}$
virtual camera 1
$\Rightarrow \mathbf{r}_2^\mathrm{T} \cdot (\mathbf{r}_3 \times \mathbf{r}_1) = 0$

Figure 5. The epipolar constraint using the corresponding virtual cameras. For an arbitrary feature point in view 1 and its matched feature point in view 2, two corresponding virtual cameras are established. The refraction rays ($\mathbf{r}_1$ and $\mathbf{r}_2$) and the ray connecting two virtual cameras' centers ($\mathbf{r}_3$) are co-planar.

in view 2, two corresponding virtual cameras are obtained. The rays after refraction ($\mathbf{r}_1$ and $\mathbf{r}_2$) and the ray connecting two virtual cameras centers ($\mathbf{r}_3$) are co-planar. In a common coordinate frame, $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ satisfy the epipolar relation

$$\mathbf{r}_2^\top \mathbf{r}_3^\wedge \mathbf{r}_1 = 0 \,. \tag{9}$$

4.2.2 Optimization Objective

We follow [5] and we optimize the relative pose and triangulated points by jointly minimising the reprojection errors and the epipolar costs

$$\mathbf{R}, \mathbf{c}, \mathbf{p}^i = \underset{\mathbf{R}, \mathbf{c}, \mathbf{p}^i}{\arg\min} \sum_{i=1}^{n} \mathrm{EC}(\mathbf{R}, \mathbf{c}, \mathbf{u}_1^i, \mathbf{u}_2^i)^2 + \sum_{i=1}^{n} \left\| \mathrm{RE}\left(\mathbf{I}, \mathbf{0}, \mathbf{p}^i, \mathbf{u}_1^i\right) \right\|_2^2 + \left\| \mathrm{RE}\left(\mathbf{R}, \mathbf{c}, \mathbf{p}^i, \mathbf{u}_2^i\right) \right\|_2^2 \,, \tag{10}$$

with $\mathrm{EC}(\mathbf{R}, \mathbf{c}, \mathbf{u}_1^i, \mathbf{u}_2^i)$ being the epipolar cost defined in the left hand side of (9) and each term $\mathrm{RE}\left(\cdot, \cdot, \mathbf{p}^i, \mathbf{u}_j^i\right)$ is the reprojection error of the i^{th} point in the j^{th} image. For relative pose estimation, we assume that the first camera coordinate frame coincides with the world frame. Thus, the camera pose for the first view is $\mathbf{I}$ and $\mathbf{0}$. Unlike [5], which computes the forward projection by solving a quartic equation, we compute it via the virtual camera coordinates. Compared with [15] that merely minimizes the reprojection error, (10) jointly minimizes the epipolar constraint and the reprojection error, offering two superiority: 1) the epipolar constraint only relies on the relative pose (no dependency on the scene) so that it helps to optimize the relative pose regardless of the 3D points. 2) less possibility of getting trapped at a local minimum as the objective is a combination of two cost surfaces.

4.3. Pose Estimation for Scenario 2

In scenario 2, illustrated in Fig. 6, we know the parameters of the refractive interface in the world coordinate frame. To get the refractive point $\mathbf{q}$ and refraction ray $\mathbf{r}$, we need to transform the back-projected ray $\mathbf{q}_\mathrm{C}$ of an image point $\mathbf{u}$ from the camera frame to the world frame, i.e. to get the camera pose $(\mathbf{R}, \mathbf{c})$. The refractive point $\mathbf{q}$ and the refraction ray $\mathbf{r}$ thus depend also on the camera pose $(\mathbf{R}, \mathbf{c})$. The dependencies of $\mathbf{q}$ and $\mathbf{r}$ on $\mathbf{R}, \mathbf{c}$ are nonlinear due to Snell's law. This makes deriving the absolute pose estimation solution for scenario 2 more complicated than for scenario 1. We give a relatively simple derivation of it with the help of an ideal world coordinate frame. As the following derivations involve multiple coordinate frames and transformations, we introduce, for clarity, new notations:

- In general, ${}^b\mathbf{R}_a$ denotes a rotation matrix from coordinate frame a to coordinate frame b. ${}^b\mathbf{c}_a$ denotes the origin of coordinate frame a in coordinate frame b. $\mathbf{R}$ denotes a rotation matrix from world coordinates to cameras coordinates (i.e. ${}^\mathrm{C}\mathbf{R}_\mathrm{W}$) and $\mathbf{c}$ denotes the location of the camera frame with respect to the world frame (i.e. ${}^\mathrm{W}\mathbf{c}_\mathrm{C}$).
- $\mathbf{a}_1^i$ represent a variable (e.g. a point, a ray, or a scalar) indexed or named by i and defined in the coordinate frame 1.

4.3.1 Absolute Pose Estimation

First, we define an ideal world coordinate frame (denoted by I) in which the refractive plane is at the origin ($\mathbf{0}$) and its normal is $\mathbf{n}_\mathrm{I} = \mathbf{e}_3$, as illustrated in Fig. 6. The rigid transformation ${}^\mathrm{W}\mathbf{R}_\mathrm{I}$ and ${}^\mathrm{W}\mathbf{c}_\mathrm{I}$ of the ideal world frame with respect to the world frame can be computed by aligning $\mathbf{n}_\mathrm{W}$ to $\mathbf{e}_3$ using ${}^\mathrm{W}\mathbf{R}_\mathrm{I}$ (computed by (4)) and moving the refractive plane so that the origin lies on it:

$${}^\mathrm{W}\mathbf{c}_\mathrm{I} = -\mathrm{d}_\mathrm{W} \mathbf{n}_\mathrm{W} \,, \tag{11}$$

where $\mathbf{n}_\mathrm{W}$ and d_W are the plane parameters in the world frame. The use of ideal world coordinates simplifies the derivation.

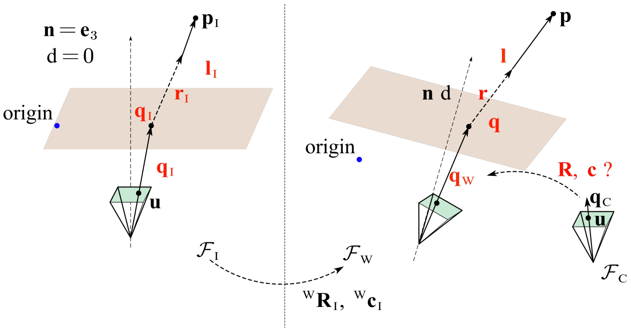


Figure 6. Configuration of scenario 2. The known variables are drawn in black color, while the unknown variables are depicted in red color. As $\mathbf{n}$ and d are only known in the world coordinate frame, the refractive point $\mathbf{q}$ and the refraction ray $\mathbf{r}$ have dependencies on the camera pose $\mathbf{R}, \mathbf{c}$ and cannot be computed without knowing them.

Let $\mathbf{K}$ be the camera intrinsic parameter matrix. The back-projected ray of an arbitrary image point $\mathbf{u}$ (in homogeneous coordinate) in the world frame is given by $\mathbf{q}_\mathrm{W} = \mathbf{R}^\top \mathbf{q}_\mathrm{C}$, where $\mathbf{q}_\mathrm{C} = \frac{\mathbf{K}^{-1}\mathbf{u}}{\|\mathbf{K}^{-1}\mathbf{u}}\|$. By further transforming it into the ideal world frame, we will get a ray $\mathbf{q}_\mathrm{I} = {}^\mathrm{W}\mathbf{R}_\mathrm{I}^\top \mathbf{q}_\mathrm{W}$. Moreover, the camera location in the ideal world frame is given as ${}^\mathrm{I}\mathbf{c}_\mathrm{C} = {}^\mathrm{W}\mathbf{R}_\mathrm{I}^\top(\mathbf{c} - {}^\mathrm{W}\mathbf{c}_\mathrm{I})$.

In the ideal world frame, computing the refractive point and the refraction ray is easily done using (1) and (2) (we can assume, without loss of generality, that ${}^\mathrm{I}\mathbf{c}_\mathrm{C}(3) < 0$, i.e. the camera is below the refractive plane)

$$\begin{aligned}\mathbf{q}_\mathrm{I} &= \begin{bmatrix} \mathbf{q}_\mathrm{I}(3)^\mathrm{I}\mathbf{c}_\mathrm{C} -^\mathrm{I} \mathbf{c}_\mathrm{C}(3)\mathbf{q}_\mathrm{I} \\ \mathbf{q}_\mathrm{I}(3) \end{bmatrix} \\ \mathbf{r}_\mathrm{I} &= \begin{bmatrix} \lambda\mathbf{q}_\mathrm{I}(1) & \lambda\mathbf{q}_\mathrm{I}(2) & \gamma & 0 \end{bmatrix}^\top ,\end{aligned} \tag{12}$$

where $\gamma = \sqrt{(1-\lambda^2) + \lambda^2\mathbf{q}_\mathrm{I}(3)^2}$. The Plücker coordinate $\mathbf{l}_\mathrm{I}$ of the line passing by $\mathbf{q}_\mathrm{I}$ and parallel to $\mathbf{r}_\mathrm{I}$ is obtained by (6). The corresponding 3D point $\mathbf{p}_\mathrm{W}$ is then transformed to the ideal world coordinate frame as $\mathbf{p}_\mathrm{I} =^\mathrm{W} \mathbf{R}_\mathrm{I}^\top(\mathbf{p}_\mathrm{W} - {}^\mathrm{W}\mathbf{c}_\mathrm{I})$.

The absolute pose estimation objective is defined in (8), the residual function is given by the left hand side of (7)

$$\mathbf{f} = \mathbf{l}_\mathrm{I}(1,2,3) \times \mathbf{p}_\mathrm{I} + \mathbf{l}_\mathrm{I}(4,5,6). \tag{13}$$

Its Jacobian matrix $\mathbf{J}_\mathbf{x} = [\mathbf{J}_\mathbf{R}, \mathbf{J}_\mathbf{c}]$ is computed in the supplementary material.

4.3.2 Relative Pose Estimation

We compute residual functions for the optimisation objective defined in (10). We again use ideal world coordinate frame defined by (11). We also obtain inhomogeneous coordinates of the refractive point and the refraction ray (i.e. $\mathbf{r}_\mathrm{I}^1$ and $\mathbf{r}_\mathrm{I}^2$) from (12). Then, we compute virtual camera coordinate following Section 3.2. $\mathbf{n}_\mathrm{I}$ is $\mathbf{e}_3$ in the ideal world coordinate frame, the resulting rotation matrix ${}^\mathrm{I}\mathbf{R}_\mathrm{V}$ is $\mathbf{I}$. Next, to find the virtual camera location ${}^\mathrm{I}\mathbf{c}_\mathrm{V}$, we need first to solve the following equation

$$\mathbf{q}_\mathrm{I} + \alpha_1\mathbf{r}_\mathrm{I} = {}^\mathrm{I}\mathbf{c}_\mathrm{C} + \alpha_2\mathbf{n}_\mathrm{I} \,. \tag{14}$$

This equation has a closed-form solution, with $\alpha_1 = \frac{{}^\mathrm{I}\mathbf{c}_\mathrm{C}(3)}{\lambda\mathbf{q}_\mathrm{I}(3)}$. Therefore, the virtual camera's location ${}^\mathrm{I}\mathbf{c}_\mathrm{V}$ is

$${}^\mathrm{I}\mathbf{c}_\mathrm{V} = \begin{bmatrix} {}^\mathrm{I}\mathbf{c}_\mathrm{C}(1) & {}^\mathrm{I}\mathbf{c}_\mathrm{C}(2) & \frac{{}^\mathrm{I}\mathbf{c}_\mathrm{C}(3)}{\lambda\mathbf{q}_\mathrm{I}(3)}\gamma \end{bmatrix}^\top \,. \tag{15}$$

Then the ray $\mathbf{r}_\mathrm{I}^3$ that connects the two virtual camera centers is $\mathbf{r}_\mathrm{I}^3 = {}^\mathrm{I}\mathbf{c}_\mathrm{V}^2 - {}^\mathrm{I}\mathbf{c}_\mathrm{V}^1$.

With this, we can now write down the corresponding residual functions.

Epipolar Cost: The residual function for the epipolar cost term is given by:

$$\mathbf{f}_\mathrm{EC} = \mathbf{r}_\mathrm{I}^{2\top}({}^\mathrm{I}\mathbf{c}_\mathrm{V}^2 -^\mathrm{I} \mathbf{c}_\mathrm{V}^1)^\wedge \mathbf{r}_\mathrm{I}^1, \tag{16}$$

where ${}^\mathrm{I}\mathbf{c}_\mathrm{V}^i$ denotes the virtual camera location for the i^th view and $\mathbf{r}_\mathrm{I}^i$ represents the refraction ray computed from the same view. The Jacobian matrix $\mathbf{J}_{\mathrm{EC}_\mathbf{x}}$ is provided in the supplementary material.

Reprojection Error: By transforming a 3D point $\mathbf{p}$ and its corresponding refraction ray $\mathbf{r}_\mathrm{I}$ to the ideal world coordinate frame, we get $\mathbf{p}_\mathrm{V} = \mathbf{p} - {}^\mathrm{I}\mathbf{c}_\mathrm{V}$ and $\mathbf{r}_\mathrm{V} = \mathbf{r}_\mathrm{I}$. The reprojection residual is:

$$\mathbf{f}_\mathrm{RE} = \begin{bmatrix} f_\mathrm{V}\frac{\mathbf{p}_\mathrm{V}(1)}{\mathbf{p}_\mathrm{V}(3)} - f_\mathrm{V}\frac{\mathbf{r}_\mathrm{V}(1)}{\mathbf{r}_\mathrm{V}(3)} \\ f_\mathrm{V}\frac{\mathbf{p}_\mathrm{V}(2)}{\mathbf{p}_\mathrm{V}(3)} - f_\mathrm{V}\frac{\mathbf{r}_\mathrm{V}(2)}{\mathbf{r}_\mathrm{V}(3)} \end{bmatrix} \,. \tag{17}$$

The Jacobian matrix $\mathbf{J}_{\mathrm{RE}_\mathbf{x}}$ is provided in the supplementary material.

Informally, we call $\mathbf{f}_\mathrm{RE} = 0$ the refractive epipolar constraint in the following. There are a few points worth of discussion: **1**) Compared with the work by [5], our formulation is simpler. Thanks to its simple form, we are able to derive the analytical Jacobian matrices which are very useful to boost the iterative refinement. **2**) Although we derive the epipolar constraint by considering the single-layer flat refraction, it can be shown that this derived epipolar constraint also holds in more general cases (e.g. multi-layer flat refraction or non-flat refractive interface) as long as the parameters of the refractive interfaces are known. **3**) The derived refractive epipolar constraint shares a similar problem with the refractive fundamental matrix proposed by [5]; it is feature-dependent. This means that, to evaluate the cost, we need to re-compute it for each feature pair. Fortunately, the computation can be easily performed in parallel.

Initialization: There is no efficient and robust method for estimating the relative pose under refraction. A solution can be to simply ignore the refraction and use the classical *5-pt* algorithm proposed by [38] to get the relative pose. Afterward, 3D points are triangulated using the refraction rays.

4.4. Optimization on Manifolds

The optimisations problems (8) and (10) are unconstrained problems on manifolds of the type $SO(3) \times \mathbb{R}^{3n}$, $n \geq 1$. We use the manifold Levenberg-Marquardt algorithm of [2] tailored to this situation. Using rectified coordinates, i.e. translating the $SO(3)$-Jacobian components to $\mathfrak{so}(3)$, the tangent space of $SO(3)$ at $\mathbf{I}$, via the matrix exponential and logarithm [34], and projecting and translating back the increment back on the manifold.

5. Experiments

In this section,we evaluate the performance of algorithms targeting scenario 2. We first compare our solutions to SOTA absolute and relative pose solutions using simulation experiments. We evaluate the performance in terms of accuracy, robustness to noise, and computational efficiency. We define the rotation error and translation error between the ground truth $\mathbf{R}_{\text{tr}}, \mathbf{t}_{\text{tr}}$ and the estimation $\hat{\mathbf{R}}, \hat{\mathbf{t}_{\text{tr}}}$ as $\epsilon_{\text{rot}} = \| \log(\mathbf{R}_{\text{tr}}^\top \hat{\mathbf{R}} - \mathbf{I}) \|_{\text{fro}}$ and $\epsilon_{\text{tran}} = \| \mathbf{t}_{\text{tr}} - \hat{\mathbf{t}} \|$. Finally, we show results on real data. All experiments run on a 2.8 GHz Intel Core 2 Duo computer. Except for the EPnP algorithm that is implemented in C++, all the other methods are implemented in Matlab.

5.1. Simulation: Absolute Pose

The algorithms for comparison include the 5pt algorithm [13], the AGW algorithm [3], and the classical EPnP algorithm [31]. Regarding the implementation, we use the open-source code provided by [13] for the 5pt algorithm, while for the EPnP algorithm, we use the corresponding function provided by the OpenCV library [16]. We re-implement the AGW algorithm for handling single layer refraction as the original implementation provided by [3] was aiming to handle the double layer refraction. The proposed method is named PO for short. We use the *8-pt* algorithm [3] for initialization.

Non-Planar Case: In the non-planar case, we create random experiments by assuming a perspective camera with a focal length of 4800, a principal point of $(960, 540)$, and no distortions. For each iteration, 100 uniformly distributed features are generated and perturbed by different levels of zero-mean Gaussian noise. We assume a flat refractive plane located at the origin with the normal being $\mathbf{n} = [0, 0.5, 1]^\top$. To ensure the plane can be viewed in front of the camera, we place the camera above the refractive interface and draw the optical axis as $\mathbf{z} = [\lambda_1, \lambda_2, -1]^\top, \lambda_1, \lambda_2 \in [-0.5, 0.5]$. The simulated refractive index is 1 (air) and 1.5 (glass) and we assume the camera is in the air.

In order to evaluate the robustness to noise, we gradually increase the standard deviation of Gaussian noise from 0 to 2 pixels. For each noise level, 100 simulations are executed. In Fig. 7, the box plots show the rotation and translation error as well as the mean runtime. As can be seen, our solution PO outperforms other SOTA solvers by showing the best robustness to noise in all situations. When compared to the other iterative solver, AGW, PO not only shows superior robustness but also at a lower computational time cost.

Planar Case: In the planar case, we use the same setup for data generation, except we force all world points to be on a common plane. Again, the noise level is varied from 0 to 2 pixels and 100 simulations are conducted per each noise level. The results are shown in Fig. 8, and PO again maintains the best robustness to noise and significantly outperforms other methods. Regarding runtime, PO is faster than AGW, but slower than the EPnP and 5pt algorithm.

5.2. Simulation: Relative Pose

To generate a synthetic dataset for evaluating relative pose solvers, we first create a random refractive interface. Then, 100 features are randomly generated in the first view, and their corresponding 3D points are randomly distributed on the rays after refraction at a distance between 5 and 10 meters. Next, we project 3D points into the second view to get the feature points. Gaussian noise is added to feature points in both views. We vary the standard deviation of the noise from 0 to 2 to evaluate the robustness. For each noise level, 50 simulations are executed.

The five-point algorithm (5pt) by [38] that ignores refraction and the method (Ichimaru) by [15] are chosen as the comparison methods. Note that we carefully re-implement the method by [15] to make it work in general. We name our method Virtual Epipolar Constraint – VEC – in the following. As the scale cannot be reliably recovered, we estimate the scale by $s = \sum_i^{N-1} \hat{d}^i / d^i$, where d (or $\hat{d}$) is the Euclidean distance between two (estimated) points.

Experimental results are shown in Fig. 9. The proposed VEC algorithm does improve the camera pose initialized by the 5pt algorithm and it shows good robustness to noise. The runtime is also unsurprisingly significantly higher than that of the 5pt algorithm. Meanwhile, compared with [15], the proposed relative pose solver works better than theirs in terms of accuracy and speed.

5.3. Real Data: Bathtub Dataset

In order to evaluate the absolute and relative pose estimation on real data we carried out an experiment on a scene

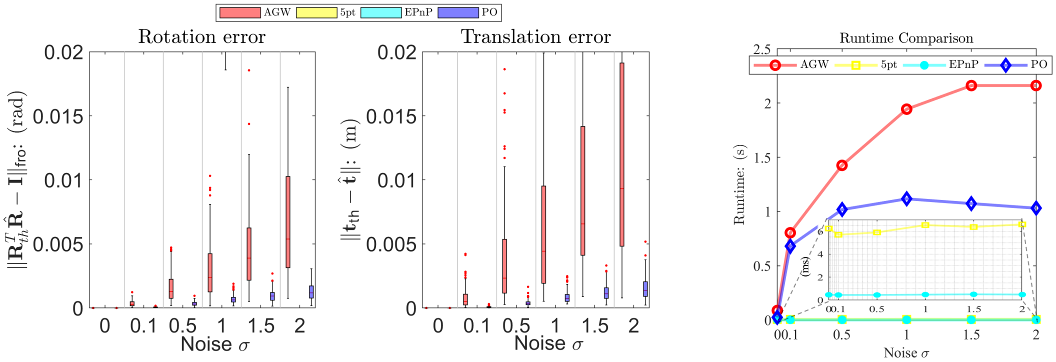


Figure 7. Comparison of absolute pose solvers for scenario 2 with respect to varying noise levels in the nonplanar case: the left box plot shows rotation error; the middle box plot shows the translation error; the right figure shows the computational time of all algorithms. Note that, for 5pt and EPnP, their errors are out of scope. A zoom-out version can be found in the supplement.

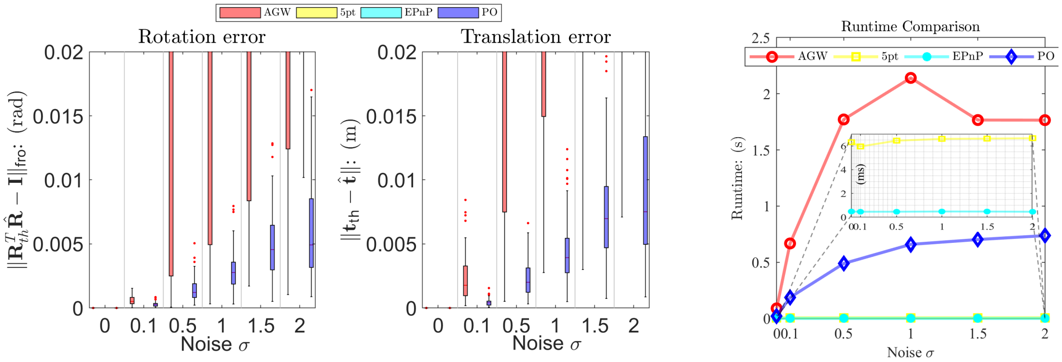


Figure 8. Comparison of absolute pose solvers for scenario 2 with respect to varying noise levels in the planar case: the left box plot shows rotation error; the middle box plot shows the translation error; the right figure shows the computational time of all algorithms. Note that, for 5pt and EPnP, their errors are out of scope. A zoom-out version can be found in the supplement.

(see Fig. 1) consisting of a water filled bathtub with a submerged checker pattern and a floating checker pattern (for estimating the refractive interface and for ground truth camera poses). We used a GoPro Hero9 Black camera with narrow lens setting (f/2.5) and 5184 × 3888 pixel resolution and collected images at 4 different views.

We use the 5pt algorithm [13] for initialization for the relative pose estimation, and the EPnP algorithm [31] for initialization for the absolute pose estimation. For comparison we estimate the reference camera pose based on the floating checker pattern using OpenCV. Fig. 11 show poses as camera frustum. We see that the estimated poses have moved away from the initialization and are very close to the ground truth camera poses.

Furthermore, we employed the estimated camera poses for a semi-dense 3D reconstruction of the submerged checker pattern, where the optical flow algorithm proposed by [27] was used to establish feature correspondences. The refractive index of water is set to 1.333. The submerged checker pattern was segmented out of raw images. In Fig. 10, we show a refractive SfM reconstruction of the submerged checker pattern. We tried to apply a classical SfM reconstruction using the COLMAP software [41]. Unfortunately, it failed to give a sparse reconstruction result, whereas our method was able to reproduce the 3D structure of the submerged checker pattern.

5.4. Real Data: Stag Beetle Dataset

For evaluating the proposed absolute and relative pose estimation applied to SfM, we carried out a real experiment where we used a NIKON D40X (focal length: 34mm, resolution: 3872 × 2592 pixels) to photograph a stag beetle enclosed in crystal resin, as shown in Fig. 1, and collected images in 30 different views. Four ArUco [11] fiducial markers were glued on the front surface of the resin to estimate the refractive interface in the world coordinate frame (the first view defines the world frame). We initialize the camera poses from the background of the scene using AliceVision [1, 36]. For pose estimation, we detected SIFT [33] features from the segmented refractive images. Feature correspondences were then established with feature matching, where erroneous matches were removed by a geometric verification. We apply the refractive bundle adjustment [20] to refine camera poses and triangulated points. Finally, we em-

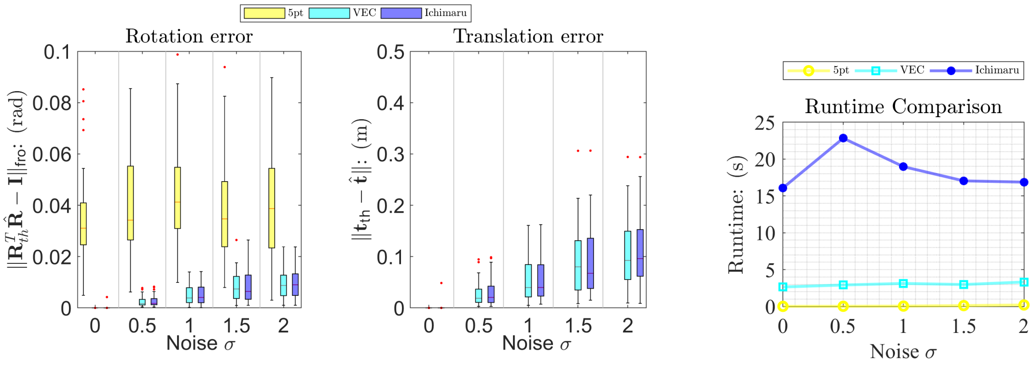
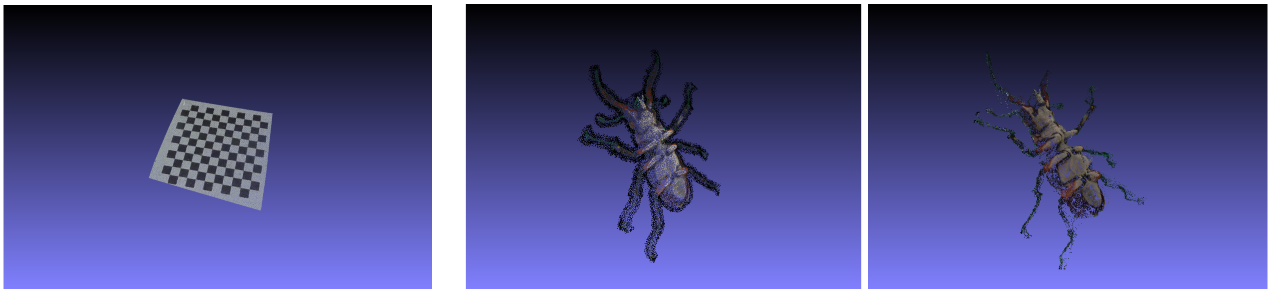


Figure 9. Comparison of relative pose solvers for scenario 2 with respect to varying noise levels: the left box plot shows rotation error; the middle box plot shows the translation error; the right figure shows the computational time of all algorithms. Note that, for 5pt, its translation errors are out of scope. A zoom-out version can be found in the supplement.



(a): Bathtub dataset.

(b): Stag Beetle Dataset.

Figure 10. SfM reconstruction result on real datasets, see Fig. 1. For (b), the left image shows SfM reconstruction based on perspective camera, while the right image shows SfM reconstruction based on refractive SfM using our method for Scenario 2.

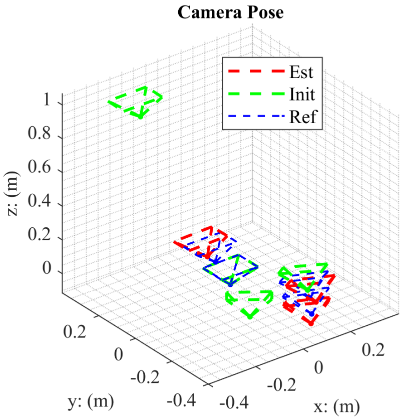


Figure 11. The ground truth camera poses (blue), initialisation (green), the estimated poses (red) using our method for scenario 2 from 4 images, see Fig. 1.

ployed the estimated camera poses for a semi-dense 3D reconstruction of the beetle, where the optical flow algorithm proposed by [27] was used to establish feature correspondences.

For comparison, we compute a classical SfM reconstruction using the COLMAP software [42].

In Fig. 10, we show a classical SfM and a refractive SfM reconstruction of the stag beetle from Fig. 1. Notice that the classical reconstruction fails to get the depth correct and produces an almost flat reconstruction, whereas our method more faithfully reproduce the 3D structure of the stag beetle (especially the legs and attachments on the body). However, the refractive reconstruction is slightly more noisy.

6. Conclusion

We have defined the theoretical framework for absolute and relative camera pose estimation under refraction with a refractive interface fixed in world coordinates (scenario 2), and we derived solutions for this scenario. We have demonstrated the superiority of our solution compared to SOTA on synthetic and real data. As iterative optimization algorithms, the proposed algorithms rely on a proper initialization. Another weakness of the proposed method is the feature matching under refraction. To our best knowledge, there is not yet a well-established initialization method nor a solution to identify erroneous matches in this scenario. Future work will consider deriving an initialization method in scenario 2, as well as how to handle the erroneous matches in pose estimation.

References

[1] AliceVision, https://alicevision.org/, March 2021. 7

[2] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 6

[3] Amit Agrawal, Srikumar Ramalingam, Yuichi Taguchi, and Visesh Chari. A theory of multi-layer flat refractive geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3346–3353. IEEE, 2012. 1, 2, 3, 6

[4] François Chadebecq, Francisco Vasconcelos, George Dwyer, René Lacher, Sébastien Ourselin, Tom Vercauteren, and Danail Stoyanov. Refractive structure-from-motion through a flat refractive interface. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5315–5323, 2017. 1, 2, 3

[5] François Chadebecq, Francisco Vasconcelos, Rene Lacher, Efthymios Maneas, Adrien Desjardins, Sébastien Ourselin, Tom Vercauteren, and Danail Stoyanov. Refractive two-view reconstruction for underwater 3d vision. *International Journal of Computer Vision*, 11 2019. 1, 2, 3, 4, 5

[6] Yao-Jen Chang and Tsuhan Chen. Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction. In *2011 International Conference on Computer Vision*, pages 351–358. IEEE, 2011. 2

[7] Visesh Chari and Peter Sturm. Multiple-view geometry of the refractive plane. In *BMVC 2009-20th British Machine Vision Conference*, pages 1–11. The British Machine Vision Association (BMVA), 2009. 1, 2, 3

[8] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–508, 2014. 2

[9] Victor Fragoso, Joseph DeGol, and Gang Hua. gdls*: Generalized pose-and-scale estimation given scale and gravity priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2210–2219, 2020. 2

[10] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 2

[11] Sergio Garrido-Jurado, Rafael Munoz-Salinas, Francisco José Madrid-Cuevas, and Rafael Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491, 2016. 7

[12] Michael D Grossberg and Shree K Nayar. A general imaging model and a method for finding its parameters. In *Proceedings of the IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 108–115. IEEE, 2001. 1

[13] Sebastian Haner and Kalle Astrom. Absolute pose for cameras under flat refractive interfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1428–1436, 2015. 1, 2, 6, 7

[14] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (dls) method for pnp. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 383–390. IEEE, 2011. 2

[15] Kazuto Ichimaru, Yuichi Taguchi, and Hiroshi Kawasaki. Unified underwater structure-from-motion. In *Proceedings - 2019 International Conference on 3D Vision, 3DV 2019*, pages 524–532, Sept. 2019. 1, 3, 4, 6

[16] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015. 6

[17] Anne Jordt. *Underwater 3D Reconstruction Based on Physical Models for Refraction and Underwater Light Propagation*. PhD thesis, 2013. 1

[18] Anne Jordt, Kevin Köser, and Reinhard Koch. Refractive 3d reconstruction on underwater images. *Methods in Oceanography*, 15:90–113, 2016. 1, 2, 3

[19] Anne Jordt-Sedlazeck and Reinhard Koch. Refractive calibration of underwater cameras. In *Proceedings of the European Conference on Computer Vision*, pages 846–859. Springer, 2012. 3

[20] Anne Jordt-Sedlazeck and Reinhard Koch. Refractive structure-from-motion on underwater images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 57–64, 2013. 1, 2, 3, 7

[21] Lai Kang, Lingda Wu, and Yee-Hong Yang. Experimental study of the influence of refraction on underwater three-dimensional reconstruction using the svp camera model. *Applied optics*, 51(31):7591–7603, 2012. 1

[22] Lai Kang, Lingda Wu, and Yee-Hong Yang. Two-view underwater structure and motion for cameras under flat refractive interfaces. In *Proceedings of the European Conference on Computer Vision*, pages 303–316. Springer, 2012. 2

[23] Laurent Kneip, Paul Furgale, and Roland Siegwart. Using multi-camera systems in robotics: Efficient solutions to the npnp problem. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3770–3776. IEEE, 2013. 2

[24] Laurent Kneip and Hongdong Li. Efficient computation of relative pose for multi-camera systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 446–453, 2014. 2

[25] Laurent Kneip, Hongdong Li, and Yongduek Seo. Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability. In *Proceedings of the European Conference on Computer Vision*, pages 127–142. Springer, 2014. 2

[26] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2969–2976, 2011. 2

[27] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *Proceedings of the European Conference on Computer Vision*, pages 471–488. Springer, 2016. 7, 8

[28] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *Proceedings of the European Conference on Computer Vision*, pages 302–315. Springer, 2008. 2

[29] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. Underwater camera calibration. In *Proceedings of the European Conference on Computer Vision*, pages 654–668. Springer, 2000. 1

[30] Gim Hee Lee, Bo Li, Marc Pollefeys, and Friedrich Fraundorfer. Minimal solutions for pose estimation of a multi-camera system. In *16th International Symposium of Robotics Research, ISRR 2013*, pages 521–538. Springer Verlag, 2016. 2

[31] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 2, 6, 7

[32] Hongdong Li, Richard Hartley, and Jae-hak Kim. A linear approach to motion estimation using generalized camera models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[33] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 7

[34] Yi Ma, Stefano Soatto, Jana Košecká, and Shankar Sastry. *An invitation to 3-D vision. From Images to Geometric Models*. Interdisciplinary Applied Mathematics. Springer, 2004. 2, 6

[35] Pedro Miraldo, Tiago Dias, and Srikumar Ramalingam. A minimal closed-form solution for multi-perspective pose estimation using points and lines. In *Proceedings of the European Conference on Computer Vision*, pages 474–490, 2018. 2

[36] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. In *Proceedings of the Asian Computer Vision Conference*, pages 257–270. Springer Berlin Heidelberg, 2012. 7

[37] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009. 2

[38] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 6

[39] David Nistér and Henrik Stewénius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, 2007. 2

[40] Robert Pless. Using many cameras as one. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–587. IEEE, 2003. 2

[41] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 7

[42] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[43] Henrik Stewenius, D Nistér, Magnus Oskarsson, and Karl Åström. Solutions to minimal generalized relative pose problems. In *OMNIVIS 2005*, 2005. 2

[44] Peter Sturm. Multi-view geometry for general camera models. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 206–212. IEEE, 2005. 1, 2

[45] Chris Sweeney, Victor Fragoso, Tobias Höllerer, and Matthew Turk. gdls: A scalable solution to the generalized pose and scale problem. In *Proceedings of the European Conference on Computer Vision*, pages 16–31. Springer, 2014. 2

[46] Gili Telem and Sagi Filin. Photogrammetric modeling of underwater environments. *ISPRS journal of photogrammetry and remote sensing*, 65(5):433–444, 2010. 3

[47] Tali Treibitz, Yoav Schechner, Clayton Kunz, and Hanumant Singh. Flat refractive geometry. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):51–65, 2011. 1

[48] Jonathan Ventura, Clemens Arth, and Vincent Lepetit. An efficient minimal solution for multi-camera motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 747–755, 2015. 2

[49] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. A minimal solution to the generalized pose-and-scale problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 422–429, 2014. 2