2017 International Conference on 3D Vision (3DV)

Generalized Pose Estimation from Line Correspondences with Known Vertical Direction

Nora Horanyi[1] and Zoltan Kato[1,2]

[1] Institute of Informatics, University of Szeged, P.O. Box 652, H-6701, Szeged, Hungary
[2] Department of Mathematics and Informatics, J. Selye University, Komarno, Slovakia,
horanyi@inf.u-szeged.hu, kato@inf.u-szeged.hu

Abstract

We propose a novel method to compute the absolute pose of a generalized camera based on straight lines, which are common in urban environment. The only assumption about the imaging model is that 3D straight lines are projected via projection planes determined by the line and camera projection directions, i.e. *correspondences are given as a 3D world line and its projection plane. Since modern cameras are frequently equipped with various location and orientation sensors, we assume that the vertical direction (*e.g. *a gravity vector) is available. Therefore we formulate the problem in terms of 4 unknowns using 3D line - projection plane correspondences which yields a closed form solution. The solution can be used as a minimal solver as well as a least squares solver without reformulation. The proposed algorithm have been evaluated on various synthetic datasets as well as on real data. Experimental results confirm state of the art performance both in terms of quality and computing time.*

1. Introduction

Pose estimation is a fundamental building block of various vision applications, *e.g.* visual odometry [22], image-based localization and navigation [12], fusion [27], and augmented reality [2]. Herein, we are interested in absolute pose estimation, which consists in determining the position and orientation of a camera with respect to a 3D world coordinate frame. Absolute pose estimation has been extensively studied yielding various formulations and solutions. Most of the approaches focus on a single perspective camera pose estimation using n 2D–3D point correspondences, known as the *Perspective n Point* (PnP) problem [18, 19, 14]. It has been widely studied for both large n as well as for the $n = 3$ minimal case (see [14] for a recent overview). Using line correspondences yields the *Perspective n Line* (PnL) problem (see [29] for a detailed overview). The minimal case of $n = 3$ line correspondences is particularly important as its solution is the basis for dealing with the general PnL problem. It has been shown in [7], that P3L leads to an 8^{th} order polynomial, which is higher than the 4^{th} polynomial of a P3P problem.

However, modern applications, especially in vision-based localization and navigation for robotics and autonomous vehicles, it is often desirable to use multi-camera systems which covers large field of views [23, 5, 17]. Not only classical image-based techniques, such as Structure from Motion (SfM) [26] provide 3D measurements of a scene, but modern range sensors (*e.g.* Lidar, Kinect) record 3D structure directly. Thus the availability of 3D data is also becoming widespread, hence methods to estimate absolute pose of a set of cameras based on 2D measurements of the 3D scene received more attention [14, 5, 16].

In this work, we deal with the generalized absolute pose estimation from 3D–2D line correspondences (also known as the gPnL problem) with known vertical direction. While several point-based methods exist [5, 14], little work has been done on using line correspondences for generalized pose. One notable work is the minimal multiview NP3L solver of Lee [16], which deals with full 6 DOF pose parameter estimation. Today, the vast majority of modern cameras, smartphones, UAVs, and camera mounted mobile platforms are equiped with a cheap and precise *inertial measurement unit* (IMU). Such devices provide the vertical direction from which one can calculate 2 rotation angles, thus reducing the free parameters from 6 to 4. The accuracy of this *up-vector* is typically between $0.5° - 0.02°$ [1]. While robust minimal solutions based on point correspondences exist for perspective cameras with known vertical direction [1, 15], none of these methods use line correspondences nor generalized non-perspective cameras.

2475-7888/17/31.00 ©2017 IEEE
DOI 10.1109/3DV.2017.00036

In this paper, we propose a novel solution to the gPnL problem with known vertical direction. The only assumption about our generalized camera [10] is that projection rays of a 3D line fall into *coplanar* subsets yielding a pencil of projection planes. Important special cases of such a camera include stereo and multiview perspective camera systems [5, 14, 16], perspective camera moving along a trajectory [5, 17, 6, 28, 30], as well as other non-perspective cameras with central omnidirectional [3, 9, 21, 24] or orthographic projection. Our algorithm can be used as a minimal gP3L solver with 3 line correspondences suitable for hypothesis testing like RANSAC [8]. Furthermore, the same algorithm can be used without reformulation for $n > 3$ lines as well as for classical single-view PnL problems. The performance and robustness of the proposed method have been evaluated on large synthetic datasets as well as on real data.

2. Generalized camera model

While the dominant imaging model in computer vision is perspective projection, recently much more complex vision sensors have been introduced in various application areas [10]. Regardless of its specific design, an imaging system maps incoming rays of light from the scene onto pixels on the detector. For a perspective imaging system, all the incoming light rays are projected directly onto to the detector plane through a single point, namely, the projection center of the perspective system. This is not true in an arbitrary system, *e.g*. a camera system could be comprised of multiple individual perspective or non-perspective imaging systems, each with its own imaging optics and image detector.

If we are not placing any restrictions on the properties of the imaging system, then all rays can be expressed using its Plücker coordinates defined in the generalized camera coordinate frame $\mathcal{C}$ [23, 5]. A Plücker coordinate is a homogeneous 6-vector composed of a pair of 3-vectors $(\mathbf{v}, \mathbf{m})$, where $\mathbf{v}$ is the unit direction vector of the line and $\mathbf{m} = \mathbf{v} \times \mathbf{p}$ is a vector whose direction is perpendicular to the plane containing the line and the origin, and $\mathbf{p}$ is an arbitrary point on the line [23, 5]. Obviously, $\mathbf{v} \cdot \mathbf{m} = 0$.

Thus any 3D point $\mathbf{X}$ in the camera frame $\mathcal{C}$ lying on the camera projection ray $(\mathbf{v}, \mathbf{m})$ can be written as

$$\mathbf{X} = \mathbf{v} \times \mathbf{m} + d\mathbf{v}, \quad d \in \mathbb{R}. \tag{1}$$

Since $\mathbf{v}$ is a unit vector, the point $(\mathbf{v} \times \mathbf{m})$ is the point on the ray closest to the origin and d is the (signed) distance from that point [23]. The above expression defines the 3D point - ray correspondence for a generalized camera. While this is sufficient to estimate the absolute pose with respect to a world coordinate frame $\mathcal{W}$, rays are usually not directly available in an imaging system, it has to be computed from recorded pixels. Following [10], we can

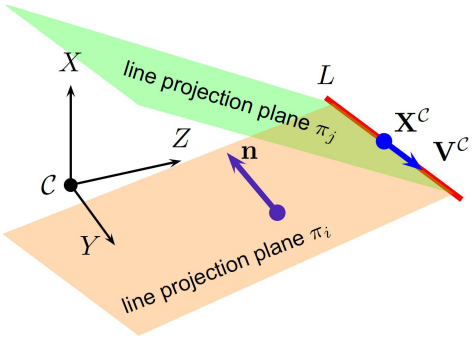

Figure 1: Projection of a 3D line L by a generalized camera.

represent the mapping of rays to pixels through so called ray pixels or *raxel*s, which are conveniently arranged on a so called *ray surface*. For example, central cameras (both perspective and non-perspective) are often represented by a unit sphere [3, 9, 21, 24]. Many imaging systems have a special ray surface, called *caustic*, which is defined as the envelope of the incoming rays (*i.e*. incoming rays are tangent to this surface). As argued by [10], caustics are the logical place to locate raxels. Note that in the perspective case, the caustic is a point (the projection center).

2.1. Line projection

While point projection of a generalized camera is governed by (1), the projection of lines becomes much more complex. For points, we assume (according to physical laws) that they are projected by *straight* lines and the image of a point will be a set of points. However lines may have a very complex projection in a generalized camera as opposed to a well defined line in the perspective case. Therefore we make an additional assumption about our camera: the projection rays of a 3D line L can be divided into *coplanar* subsets yielding a pencil of projection planes π_i^L with L being the axis of the pencil (see Fig. 1). For such a given projection plane π_i^L, one image of the line L becomes a general curve on the ray surface determined by the intersection of π_i^L with the ray surface. Common examples include multiview central cameras (see Sec. 3.2), or linear pushbroom imaging which may be thought of as a projective image in one direction and an orthographic image in the other direction.

3. Generalized absolute pose

Herein, 3D lines will be represented by their unit direction vector $\mathbf{v}$ and by an arbitrary point $\mathbf{X}$ on it: $L = (\mathbf{v}, \mathbf{X})$. The projection planes π_i^L are given by their unit normal $\mathbf{n}_i$ and the signed distance d_i from the origin, which is also known as the Hessian normal form: $\pi_i^L = (\mathbf{n}_i, d_i)$. The sign of d_i determines the side of the plane on which the origin is located. If $d_i > 0$, it is in the half-space determined

by the normal direction $\mathbf{n}_i$, and if $d_i < 0$, it is in the other half-space. Note that the (signed) point-plane distance from a homogeneous 3D point $\mathbf{X} = (X_1, X_2, X_3, 1)$ to a plane π_i^L is given by $\pi_i^L \cdot \mathbf{X}$, which should be 0 for points on the plane. If the point $\mathbf{X}$ is in the half-space determined by the normal direction $\mathbf{n}_i$, then the distance is positive; if it is in the other half-space, then the distance is negative.

Since L lies on π_i^L, its direction vector $\mathbf{v}$ is perpendicular to $\mathbf{n}_i$:

$$\forall i : \mathbf{n}_i \cdot \mathbf{v}^{\mathcal{C}} = \mathbf{n}_i \cdot \mathbf{R}\mathbf{v} = 0 \tag{2}$$

where $\mathbf{R}$ is the rotation matrix from the world $\mathcal{W}$ to the camera frame $\mathcal{C}$ and $\mathbf{v}^{\mathcal{C}}$ denotes the unit direction vector of L in the camera frame $\mathcal{C}$. Furthermore, the point $\mathbf{X}$ on line L also lies on π_i^L, hence

$$\forall i : \pi_i^L \cdot (\mathbf{X}^{\mathcal{C}}, 1) = \mathbf{n}_i \cdot (\mathbf{R}\mathbf{X} + \mathbf{t}) + d_i = 0 \tag{3}$$

where $\mathbf{t}$ is the translation from the world $\mathcal{W}$ to the camera $\mathcal{C}$ frame and $\mathbf{X}^{\mathcal{C}}$ denotes the point on L in the camera coordinate system $\mathcal{C}$.

The absolute pose of our generalized camera is defined as the rigid transformation $(\mathbf{R}, \mathbf{t})$ acting between $\mathcal{W}$ and $\mathcal{C}$, and the equations (2)–(3) provide constraints for computing the pose using 3D line and projection plane correspondences. Note that both equations have a geometric meaning: (2) expresses the cos of the angle between the plane normal and the line direction; while (3) gives the signed distance of the point on the line and its projection plane. Hence minimizing the squared error of these equations would actually minimize the geometric error.

3.1. Known vertical direction

Before formulating our solution, let us see the parameterization of $\mathbf{R}$ when the vertical direction is available. This information is typically obtained from an IMU (Inertial Measurement Unit), which consists of accelerometers capable to measure the Earth's gravity center. The accuracy of an IMU's *up-vector* is typically between $0.5° - 0.02°$ [1]. However, similar information can be obtained from a calibrated image by detecting a vanishing point (in man-made environment, one can get the vertical vanishing point) [11]. In fact, the knowledge of any direction would lead to a similar formulation of the problem.

Assuming that the camera coordinate system $\mathcal{C}$ is a standard right handed system with the X axis pointing *up* (see Fig. 1), the world unit vector $(1, 0, 0)^\top$ is known in the camera coordinate frame $\mathcal{C}$. Given this *up-vector*, we can compute the rotation $\mathbf{R}_v$ around Y and Z axes, which aligns the world X axis with the camera X axis, thus the only unknown is the rotation $\mathbf{R}_X$ around the vertical axis yielding the following factorization of the $\mathcal{W} \to \mathcal{C}$ rotation $\mathbf{R}$:

$$\mathbf{R} = \mathbf{R}_v \mathbf{R}_X = \mathbf{R}_v \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{4}$$

Unfortunately, the above factorization gives only the cos and sin of the rotation angle which, when plugged into (2)–(3), yields trigonometric equations in α but linear in $\mathbf{t}$. In order to eliminate $\sin(\alpha)$ and $\cos(\alpha)$, we can use the substitution $q = \tan(\alpha/2)$ [15, 1], for which $\cos(\alpha) = (1 - q^2)/(1 + q^2)$ and $\sin(\alpha) = 2q/(1 + q^2)$. Therefore

$$(1 + q^2)\mathbf{R}_X(q) = \begin{bmatrix} 1 + q^2 & 0 & 0 \\ 0 & 1 - q^2 & -2q \\ 0 & 2q & 1 - q^2 \end{bmatrix}. \tag{5}$$

Substituting $\mathbf{R}_X(q)$ into (2) yields a quadratic equation in the single unknown q:

$$\begin{aligned} \forall i : \quad & a_i q^2 + b_i q + c_i = 0, \text{ with} \\ a_i &= (\mathbf{n}_i^\top \mathbf{R}_v) \begin{bmatrix} v_1 \\ -v_2 \\ -v_3 \end{bmatrix} \\ b_i &= 2(v_2(\mathbf{n}_i^\top \mathbf{R}_v^1) - v_3(\mathbf{n}_i^\top \mathbf{R}_v^2)) \\ c_i &= (\mathbf{n}_i^\top \mathbf{R}_v)\mathbf{v}, \end{aligned} \tag{6}$$

where $\mathbf{v} = (v_1, v_2, v_3)^\top$ is the unit direction vector of L, while $\mathbf{R}_v^1$ and $\mathbf{R}_v^2$ denote the first and second column vector of $\mathbf{R}_v$, respectively. Once a solution is obtained for $\mathbf{R}_X$, it can be substituted into (3) yielding the following linear equation in $\mathbf{t} = (t_1, t_2, t_3)^\top$:

$$\begin{aligned} \mathbf{n}_i^\top (\mathbf{R}_v \mathbf{R}_X \mathbf{X}) + \mathbf{n}_i^\top \mathbf{t} + d_i &= 0 \\ n_{i,1} t_1 + n_{i,2} t_2 + n_{i,3} t_3 + \mathbf{n}_i^\top (\mathbf{R}\mathbf{X}) + d_i &= 0 \\ \begin{bmatrix} \mathbf{n}_i^\top & \mathbf{n}_i^\top (\mathbf{R}\mathbf{X}) + d_i \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ 1 \end{bmatrix} &= 0 \end{aligned} \tag{7}$$

where $\mathbf{X}$ is a point on the 3D line L and $\mathbf{n}_i$ is one projection plane of L in the generalized camera. Given a set of 3D lines and their projection planes, each such pair provides one linear equation of the form (7) yielding a homogeneous linear system in $\mathbf{t}$ whose matrix has rows $[\mathbf{n}_i^\top, \mathbf{n}_i^\top(\mathbf{R}\mathbf{X}) + d_i]$.

3.1.1 Number of correspondences

For each 3D line L and each corresponding projection plane π_i^L, we get one quadratic equation of the form (6) and one linear equation of the form (7). Thus having N_L distinct projections generates N_L equations for L. The total number of equations is thus $n = \sum_{\ell=1}^{M} N_\ell$ where M is the total number of 3D lines visible (*i.e.* having at least one projection plane) in our generalized camera.

The minimal case of the linear system (7) consists of three pairs of L and π^L, each one providing one equation in $\mathbf{t}$, which is easily solved. Although the minimal case for (6) would be only one pair of L and π^L, we always need 3 pairs of (L, π^L) because of $\mathbf{t}$. Therefore we only address the least squares solution of (6). We note, however, that

based on one (L, π^L) pair, the rotation parameter q could be obtained as a direct solution of the quadratic equation (6). The reason why we not proceed in this way is because the least squares formulation also leads a direct solution, hence computationally it is close to this minimal solution, while the additional constraints ensure an increased numerical stability. As a consequence, the formulation of our method is identical for the minimal as well as least squares cases. Minimal solvers are typically employed for robust pose estimation within RANSAC [8] in order to maximize the probability of picking an all-inlier sample, hence reducing the number of iterations.

3.1.2 Efficient solution: gPnLup

In the non-minimal case, both (6) and (7) becomes overdetermined and can be solved in the least squares sense. Let us emphasize, that the least squares solution minimizes the *geometric error* of the solution as (6) expresses the cos of the angle between the plane normal and the line direction; while (7) gives the signed distance of the point $\mathbf{X}$ on the line and its projection plane. The squared error of (6) becomes a quartic function in q:

$$\sum_{i=1}^{n} \left(a_i^2 q^4 + 2a_i b_i q^3 + (2a_i c_i + b_i^2) q^2 + 2b_i c_i q + c_i^2\right), \tag{8}$$

whose minima is found by computing the roots of its derivative

$$\sum_{i=1}^{n} \left(4a_i^2 q^3 + 6a_i b_i q^2 + (4a_i c_i + 2b_i^2) q + 2b_i c_i\right) \tag{9}$$

The roots of this third order polynomial are computed in a closed form. In general, there are maximum 3 roots, at least one of them being real. In case of multiple solutions, each real root is back-substituted into (7) to compute the corresponding $\mathbf{t}$. The final solution is then selected by checking whether lines are consistently placed w.r.t. the camera system, or by evaluating the reprojection error of each solution and choosing the one with minimal error.

3.2. Multiview central cameras

One common example of a generalized camera is a set of central cameras [14, 5, 23, 17]. Let us have a closer look at this important special case, when 3D lines are viewed by N central cameras. Note that a central camera may or may not be perspective! Even when a camera has a single effective viewpoint, its projection model may include non-linear distortions, like in the case of central omnidirectional cameras [3, 9, 13, 25, 27]. Herein, we will consider an arbitrary mixture of perspective and non-perspective central cameras and derive equations for (6) and (7) for this important special case.

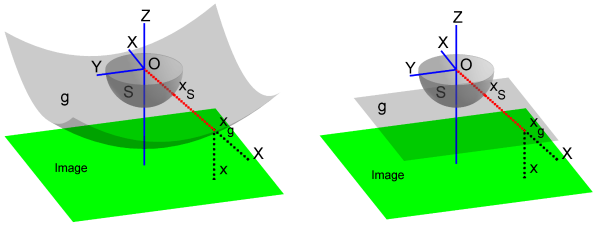

Figure 2: Spherical representation of central projection cameras.

A unified model for central omnidirectional cameras was proposed by Geyer and Daniilidis [9], which represents central panoramic cameras as a projection onto the surface of a unit sphere $\mathcal{S}$. The camera coordinate system is in the center of $\mathcal{S}$, and the Z axis is the optical axis of the camera which intersects the image plane in the *principal point*. This formalism has been adopted and models for the internal projection function have been proposed by Micusik [21, 20] and subsequently by Scaramuzza [25] who derived a general polynomial form $g(\|\mathbf{x}\|) = a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4$ which has 4 parameters representing the internal calibration parameters (a_0, a_2, a_3, a_4) of the camera (only 4 parameters as a_1 is always 0 [25]). Thus the nonlinear (but symmetric) distortion of central omnidirectional optics is represented by placing this rotationally symmetric g surface between the image plane and the unit sphere $\mathcal{S}$ [25] (see Fig. 2). Knowing the internal calibration of the camera allows us to work directly with spherical image points $\mathbf{x}_\mathcal{S} \in \mathcal{S}$ using the bijective mapping of image points $\mathbf{x} \mapsto \mathbf{x}_\mathcal{S}$ composed of 1) lifting the image point $\mathbf{x}$ onto the g surface by an orthographic projection

$$\mathbf{x}_g = \begin{bmatrix} \mathbf{x} \\ a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4 \end{bmatrix} \tag{10}$$

and then 2) centrally projecting the lifted point $\mathbf{x}_g$ onto the surface of the unit sphere $\mathcal{S}$:

$$\mathbf{x}_\mathcal{S} = \frac{\mathbf{x}_g}{\|\mathbf{x}_g\|} \tag{11}$$

Similarly, the image points of a perspective camera can be represented on $\mathcal{S}$ by the bijective mapping $\mathbf{x} \mapsto \mathbf{x}_\mathcal{S}$: $\mathbf{x}_K = \mathbf{K}^{-1}\mathbf{x}$ and $\mathbf{x}_\mathcal{S} = \mathbf{x}_K / \|\mathbf{x}_K\|$ (see Fig. 2). Thus the projection of a calibrated central camera is fully described by means of unit vectors $\mathbf{x}_\mathcal{S}$ in the half space of $\mathbb{R}^3$. A 3D world point $\mathbf{X}$ is projected into $\mathbf{x}_\mathcal{S} \in \mathcal{S}$ by a simple central projection taking into account the pose:

$$\mathbf{x}_\mathcal{S} = \frac{\mathbf{R}\mathbf{X} + \mathbf{t}}{\|\mathbf{R}\mathbf{X} + \mathbf{t}\|} \tag{12}$$

Following the line projection model outlined in Section 2.1, a 3D line L is centrally projected by a single pro-

jection plane $\pi_L = (\mathbf{n}, d)$ onto the surface $\mathcal{S}$. Since the camera projection center is also on π_L, d becomes zero and thus π_L is uniquely determined by its unit normal $\mathbf{n}$. The image of L is the intersection of the *ray surface* $\mathcal{S}$ and π_L, which is a *great circle*, while a particular line segment becomes a *great circle segment* on the unit sphere $\mathcal{S}$. When we have N central cameras, a 3D line L has up to N images, one in each camera. These cameras may be rigidly assembled into a multi-camera system or they might originate from a single camera moving along a trajectory [23, 5, 17, 2] – in either case, they form a generalized camera with known relative poses $\mathcal{C} \rightarrow \mathcal{C}_i$ with respect to the common camera coordinate frame $\mathcal{C}$. This is obtained by *e.g.* calibration of the multi-camera system [23] or by tracking the pose of a moving camera using visual-inertial-odometry [12, 5] or visual-odometry [22, 5]. Thus individual cameras are related to $\mathcal{C}$ via $(\mathbf{R}^i, \mathbf{t}^i) : \mathcal{C} \rightarrow \mathcal{C}_i$ and the projection plane $\pi^{\mathcal{C}^i} = (\mathbf{n}^{\mathcal{C}^i}, 0)$ of L in camera $\mathcal{C}^i$ is given in the generalized camera frame $\mathcal{C}$ as

$$\pi_i^L = (\mathbf{n}_i, d_i) \text{ with } \mathbf{n}_i = \mathbf{R}^{i\top}\mathbf{n}^{\mathcal{C}^i} \text{ and } d_i = \mathbf{n}^{\mathcal{C}^i} \cdot \mathbf{t}^i \quad (13)$$

Substituting the above expressions for $\mathbf{n}_i$ and d_i into (6)–(7) gives the equations for multiview central cameras.

4. Experimental Results

For the quantitative evaluation of our generalized pose estimation algorithm with line correspondences, we generated various benchmark datasets of 3D–2D line pairs. Each dataset has 1000 samples. 3D lines were generated by placing three 2D planes in the 3D Euclidean space and about 10 lines were placed on each of these planes, whose size was normalized into a 1m^3 cube. Then we applied a random translation of $0 - 1$unit and rotation of $0°, \ldots, 45°$ around the Z axis and $20°, \ldots, 60°$ around the vertical X axis to place the planes. All the parameters were inspired by common urban structural properties, in which environment the real data experiments were performed too.

The synthetic 2D images of the 3D lines were generated with omnidirectional cameras and perspective cameras as well. We applied the same rotation range of $-20°, \ldots, 20°$ around all three axis and random displacement of 3 units on every camera systems. The known intrinsic parameters of the perspective cameras are $\mathbf{K}$: focal length $f_x = 846.1251$, $f_y = 846.1424$, and the principal point $\mathbf{o}$ which was set to the center of the 2376×1584 image plane. While in case of the omnidirectional camera we used a virtual camera to generate omnidirectional images with size of 3.6MPx. The camera parameters were taken from a real 8mm Fish-eye camera calibration, calibrated with the calibration toolbox of Scaramuzza [25]. Separate datasets were generated by the combination of the perspective and omnidirectional cameras into a camera system.

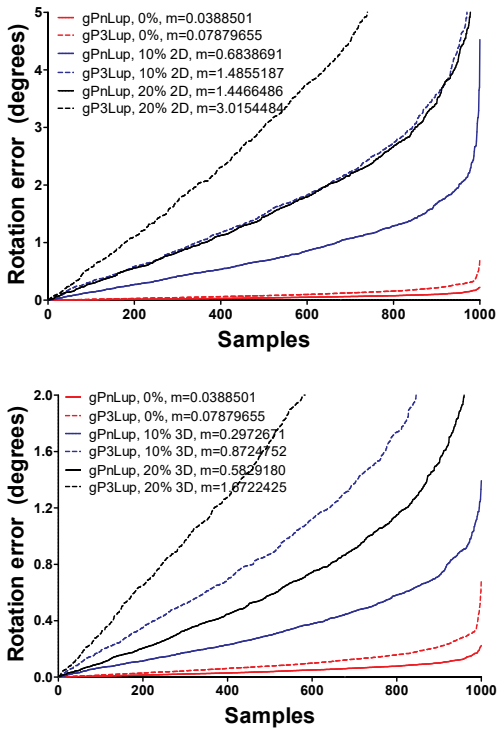

Figure 3: Comparison of our method in case of varying line numbers and varying noise levels (10% and 20% noise level,m:median error value). The upper plot indicates the efficiency of our minimal ($n = 3$) and least square solutions in case of 2D noise, while the plot below shows the results in case of 3D noise. In both of these cases we used one omnidirectional and two perspective cameras in our camera system.

In case of 3 camera systems, we used three different constructions which contains 0,1,2 or 3 omnidirectional cameras. For this setup, datasets were generated with random translation 0.4 and random relative rotation between the cameras around the Y and Z axis in range of $-5°, \ldots, 5°$, and around the X axis $15°, \ldots, 25°$.

The evaluation of the algorithms were done in two scenario: either all of the 3D–2D line pairs are used (about 30 line pairs per sample - this will be denoted by n) or only the minimum number line pairs are used (3 line pairs - this will be denoted by 3). We also compare our results with state of the art methods. Since to the best of our knowledge, there is no prior method for generalized pose estimation from line correspondences and known vertical direction, we compare our method with the point-based non-perspective UPnP [14] of Kneip *et al.*; and the line-based non-perspective minimal solver NP3L [16] of Lee.

First, we evaluate the sensitivity of our algorithms with n lines for the composition of the camera systems. Fig. 6

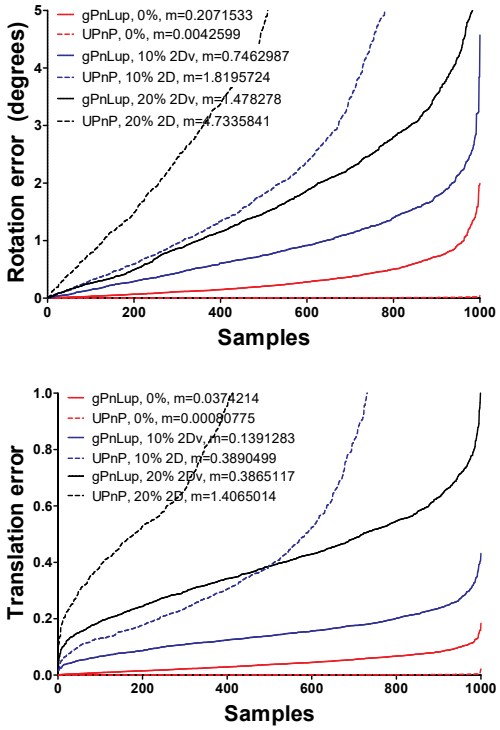

Figure 4: Comparison of the UPnP method with our method in case of varying 2D noise levels (2D: 10% and 20% 2D noise, 2Dv: 10% and 20% 2D noise with 5° vertical noise, m:median error value). The upper plot presents the rotation errors w.r.t. different 2D noise levels, while the plot lower shows translation errors. In both of these cases we used one omnidirectional and two perspective cameras in our camera system.

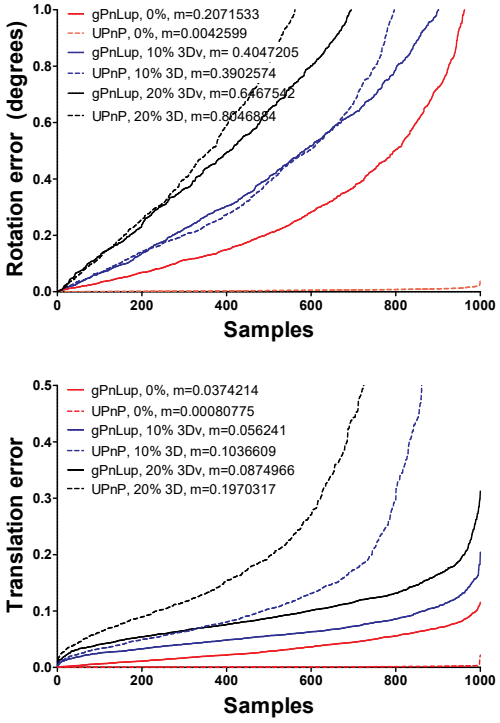

Figure 5: Comparison of the UPnP method with our method in case of varying 3D noise levels (3D: 10% and 20% 3D noise, 3Dv: 10% and 20% 3D noise with 5° vertical noise, m:median error value). The upper plot presents the rotation errors w.r.t. different 3D noise levels, while the plot lower shows translation errors. In both of these cases we used one omnidirectional and two perspective cameras in our camera system.

shows, that for n lines our solver is slightly more accurate when we have less omnidirectional camera in our system due to their effective lower resolution. However, overall the method performs quite well independently of the type of construction, having a median rotation error less than $0.0501°$ in all samples. Hereafter we use the 1 omnidirectional + 2 perspective camera configuration for our comparisons.

Next, we compare the performance in the minimal and n-line cases. Fig. 3 shows the rotational and translation error in case of n lines as well as 3 lines. Obviously, the algorithms perform better for n lines, but overall the estimates are quite accurate both in case of 2D and 3D noise.

4.1. Noise Resilience

In order to evaluate the sensitivity of our algorithms to line measurement noise, we add random noise to the generated test cases in the following way: The 2D lines are corrupted with additive random noise on one endpoint of the line and the direction vector of the line. The amount of noise is 10% and 20%, meaning that a random number is added to each coordinate up to the specified percentage of the actual coordinate value.

In Fig. 7 we show the evaluation of our algorithm in case of $1°$, $5°$ and $10°$ vertical direction noise levels. To have a fair comparison with the other methods, later we added a $\pm 5°$ random noise to the vertical direction (this is above the typical noise level of a low quality IMU), and then run our algorithms on the 10% and 20% noisy datasets both in case of 2D and 3D noise types. The comparative results of these experiments are shown in Fig. 4 and Fig. 5, where we show error plots compared with UPnP [14]. UPnP is slightly more accurate for noiseless input, but it is consistently outperformed by our methods in every noisy case. The translation errors show us the same tendency, in the noisy cases our algorithm performs better. We should note here that because the rotation and translation is decoupled, any error in the rotation is directly propagated into the linear system of

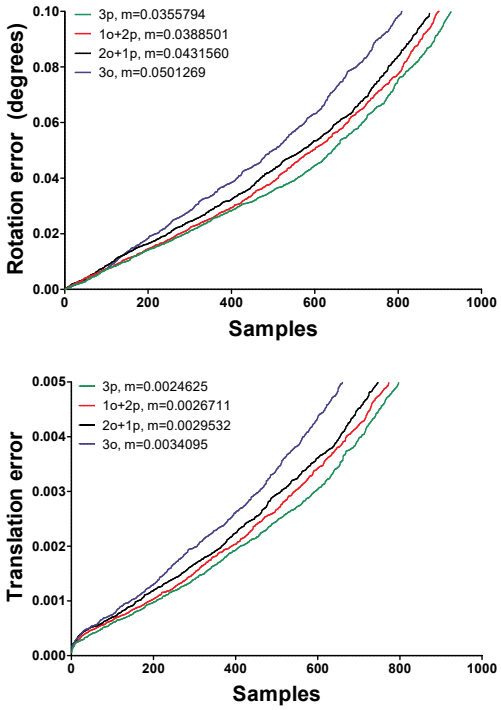

Figure 6: Efficiency of our method in case of different camera system configurations (o:omnidirectional camera, p:perspective cameras, m:median error value). The upper plot presents the rotation errors w.r.t. different camera system configurations, while the plot below shows translation errors.

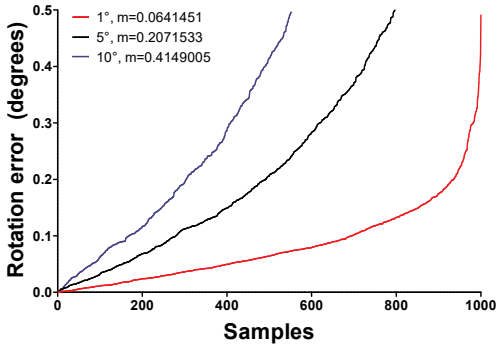

Figure 7: Comparison of the effect of various vertical direction errors in case of 1°, 5° and 10° using one omnidirectional and two perspective cameras.

the translation. Note as well, that in the maximal case when we have 3 cameras, UPnP has more than 150 point pairs to work with (we used 2 points per line), which is double than the number of correspondences for our algorithm. Finally, the typical runtime of our method was 9.8ms, while it was 8.8ms for UPnP [14] in case of n lines. CPU time of our method is slightly higher than UPnP [14], however our algorithm was implemented in MATLAB, while UPnP is implemented in C++.

	Fig. 8	Fig. 9	
	gPnLup	gPnLup	NP3L [16]
Rotation error (deg)	1.1972	1.0216	4.5029
Translation error	0.8797	0.9088	3.0037
Forward projection error(m)	0.2407	0.2904	0.5901

Table 1: Comparison of the maximal rotational, translational and forward projection error of various methods on the real data.

4.2. Real Data

In Fig. 8 and Fig. 9, we show our results on a real dataset. 2D perspective images were captured with a Canon 5D DSLR camera, the 2D omnidirectional images were taken with a Canon EF 8-15mm f4L fisheye lens, and the 3D point cloud was captured with a Riegl VZ400 Lidar scanner with an angular resolution of 0.05°. The ground truth pose was calculated using special Lidar markers placed on the building facades. The 3D location of these markers are read by the Lidar scanner, and their corresponding 2D locations are manually selected in each camera image. The role of these markers is two-fold: First, using these high quality 3D-2D point correspondences, UPnP [14] was used to calculate the reference pose for each camera image. Second, given an estimated pose, we can compute the forward projection of the 2D marker points onto the 3D surfaces and compute the (metric) error in 3D space. For the reference pose, the maximum of this *forward projection* error was 0.1205m, while the mean was 0.0677m. We detected 2D lines on the perspective images using the OpenCV LSD detector, while on the omnidirectional images we used the automatic line extraction toolbox of Bermudez [4]. 3D lines were extracted in a similar way on Riegl's own 2D RGB images used for colorizing the Lidar scan.

In the first test case (Fig. 8), we have a calibrated camera system consisting of 3 perspective and 1 omnidirectional cameras. The evaluation of the pose obtained by our gPnLup algorithm are shown in Table 1. To our best knowledge there is no existing method which works for n lines and general cameras. Hence we prepared another test case where we compare our results with the NP3L minimal solver of Lee [16]. The MATLAB implementation of NP3L provided by the author of [16] works only with three perspective cameras and a total of three line correspondences, hence in Fig. 9 we show results for such a camera system. As it is shown in Table 1, our algorithm outperforms NP3L in spite of the fact, that the input vertical direction for our

Figure 8: Lidar laser scan for testing our pose estimation algorithms with a 3-perspective-1-omnidirectional multi-view camera system. The extracted 2D lines are shown on the 2D images, while on the Lidar scan (middle) red dots are the estimated positions and green dots are the real location of the markers in metric 3D space.

Figure 9: Lidar laser scan for testing our pose estimation algorithms with a 3-perspective camera system. The extracted 2D lines are shown on the 2D images (left). On the Lidar scan (right), red dots are the estimated positions of our minimal solver, blue dots are the estimated positions of NP3L [16], and green dots are the real location of the markers in the 3D metric space.

algorithm had a 1.198° deviation from the ground truth. It is thus fair to say that gP3Lup provides state-of-the-art estimates under real conditions.

5. Conclusions

We proposed a direct least squares solutions to the gPnL problem from line correspondences with known vertical direction. The only assumption about our generalized camera is that 3D lines project through projection planes. Many practically important camera setup corresponds to this model: stereo and multiview central camera systems composed of perspective and non-perspective (*e.g.* omnidirectional) cameras, or a camera (system) moving along a trajectory. The method can be used as a minimal solver (*e.g.* within RANSAC) as well as a general least squares solver without reformulation. The proposed method have been evaluated on synthetic and real datasets. Comparative tests confirm state of the art performance both in terms of quality and computing time.

Acknowledgement

This work was partially supported by the NKFI-6 fund through project K120366;the Agence Universitaire de la Francophonie (AUF) and the Romanian Institute for Atomic Physics (IFA), through the AUF-RO project NETASSIST; "Integrated program for training new generation of scientists in the fields of computer science", no EFOP-3.6.3-VEKOP-16-2017-0002; the Research & Development Operational Programme for the project "Modernization and Improvement of Technical Infrastructure for Research and Development of J. Selye University in the Fields of Nanotechnology and Intelligent Space", ITMS 26210120042, co-funded by the European Regional Development Fund.

References

[1] C. Albl, Z. Kukelova, and T. Pajdla. Rolling shutter absolute pose problem with known vertical direction. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3355–3363, Las Vegas, NV, USA, June 2016. 1, 3

[2] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In *Proceedings of International Symposium on Mixed and Augmented Reality*, pages 37–46, Basel, Switzerland, Oct. 2011. IEEE Computer Society. 1, 5

[3] S. Baker and S. K. Nayar. A Theory of Single-Viewpoint Catadioptric Image Formation. *International Journal of Computer Vision*, 35(2):175–196, 1999. 2, 4

[4] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero. Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. *International Journal of Computer Vision*, 114(1):16–37, Aug. 2015. 7

[5] F. Camposeco, T. Sattler, and M. Pollefeys. Minimal solvers for generalized pose and scale estimation from two rays and one point. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Proceedings of European Conference Computer Vision*, volume 9909 of *Lecture Notes in Computer Science*, pages 202–218, Amsterdam, The Netherlands, Oct. 2016. Springer. 1, 2, 4, 5

[6] M. K. Chandraker, J. Lim, and D. J. Kriegman. Moving in stereo: Efficient structure and motion using lines. In *International Conference on Computer Vision*, pages 1741–1748, Kyoto, Japan, Oct. 2009. 2

[7] H. Chen. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, 1991. 1

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2, 4

[9] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems. In *European Conference on Computer Vision*, pages 445–462, Dublin, Ireland, June 2000. 2, 4

[10] M. D. Grossberg and S. Nayar. A general imaging model and a method for finding its parameters. In *International Conference on Computer Vision*, pages 108–115, 2001. 2

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2004. 3

[12] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-IMU-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014. 1, 5

[13] J. Kannala and S. S. Brandt. A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006. 4

[14] L. Kneip, H. Li, and Y. Seo. UPnP: an optimal O(n) solution to the absolute pose problem with universal applicability. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Proceedings of European Conference Computer Vision, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 127–142, Zurich, Switzerland, Sept. 2014. Springer. 1, 2, 4, 6, 7

[15] Z. Kukelova, M. Bujnak, and T. Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *Proceedings of Asian Conference on Computer Vision, Part II*, volume 6493 of *LNCS*, pages 216–229, Queenstown, New Zealand, Nov. 2010. Springer. 1, 3

[16] G. H. Lee. A minimal solution for non-perspective pose estimation from line correspondences. In *Proceedings of European Conference on Computer Vision*, pages 170–185, Amsterdam, The Netherlands, Oct. 2016. Springer. 1, 2, 6, 7, 8

[17] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, Portland, OR, USA, June 2013. 1, 2, 4, 5

[18] V. Lepetit, F.Moreno-Noguer, and P.Fua. EPnP: an accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2), 2009. 1

[19] S. Li, C. Xu, and M. Xie. A robust O(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1444–1450, 2012. 1

[20] B. Mičušík. *Two-View Geometry of Omnidirectional Cameras*. Phd thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic, June 2004. 4

[21] B. Mičušík and T. Pajdla. Para-catadioptric Camera Autocalibration from Epipolar Geometry. In *Asian Conference on Computer Vision*, volume 2, pages 748–753, Seoul, Korea South, January 2004. 2, 4

[22] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition*, volume 1, pages 1–8, Washington, DC, USA, June 2004. IEEE. 1, 5

[23] R. Pless. Using many cameras as one. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2003. 1, 2, 4, 5

[24] D. Scaramuzza, A. Martinelli, and R. Siegwart. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In *International Conference on Computer Vision Systems*, pages 45–51, Washington, USA, January 2006. 2

[25] D. Scaramuzza, A. Martinelli, and R. Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *International Conference on Intelligent Robots*, pages 5695–5701, Bejing, October 2006. 4, 5

[26] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *ACM SIGGRAPH*, pages 835–846, Boston, Massachusetts, 2006. ACM. 1

[27] L. Tamas, R. Frohlich, and Z. Kato. Relative pose estimation and fusion of omnidirectional and lidar cameras. In L. de Agapito, M. M. Bronstein, and C. Rother, editors, *Proceedings of the ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, vol-

ume 8926 of *Lecture Notes in Computer Science*, pages 640–651, Zurich, Switzerland, Sept. 2014. Springer. 1, 4

[28] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, Nov. 1995. 2

[29] C. Xu, L. Zhang, L. Cheng, and R. Koch. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1

[30] L. Zhang and R. Koch. Hand-held monocular SLAM based on line segments. In *Proceedings of the Irish Machine Vision and Image Processing Conference*, pages 7–14, Dublin, Ireland, 2011. IEEE Computer Society. 2