Globally Optimal Pose Estimation from Line Correspondences

Faraz M. Mirzaei and Stergios I. Roumeliotis

Abstract— **Correspondences between 2D lines in an image and 3D lines in the surrounding environment can be exploited to determine the camera's position and attitude (pose). In this paper, we introduce a novel approach to estimate the camera's pose by directly solving the corresponding least-squares problem algebraically. Specifically, the optimality conditions of the least-squares problem form a system of polynomial equations, which we efficiently solve through the eigendecomposition of a so-called multiplication matrix. Contrary to existing methods, the proposed algorithm (i) is guaranteed to find the globally optimal estimate in the least-squares sense, (ii) does not require initialization, and (iii) has computational cost only linear in the number of measurements. The superior performance of the proposed algorithm compared to previous approaches is demonstrated through extensive simulations and experiments.**

I. Introduction

Determining a camera's position and attitude (pose) from known 3D lines and their projections (corresponding 2D lines) in an image has numerous applications in robot localization, computer vision, and augmented reality. While several algorithms exist for *algebraically* determining a camera's pose based on line correspondences [1], [2], [3], they are specifically designed for *noise-free* scenarios where the measurement constraints are exactly satisfied. In the presence of noise, the camera pose computed by these algorithms may become unreliable and inaccurate, since the impact of noise is not explicitly modeled.

On the other hand, many iterative algorithms exist which account for measurement noise by formulating the camera pose estimation as a *nonlinear* least-squares problem [4], [5], [6]. However, these methods do not provide any guarantee of global optimality since the iterative minimization of least-squares cost functions only converges to a stationary point. In fact, in the absence of accurate initialization, these approaches often converge to a point far from the true sensor pose. One workaround is to use the output of an algebraic method for initializing an iterative least-squares algorithm. This, however, inherits the unreliability of existing algebraic methods, and does not ensure convergence to the globally optimal solution.

To address these issues, we introduce a novel approach that *directly* computes the *global minimum* of the nonlinear least-squares cost for the camera's attitude in one step using tools from algebraic geometry. Moreover, we show that once the sensor's attitude is found, its position can be readily computed from the measurements by means of ordinary least squares. Specifically, we first address the more challenging task of attitude determination by considering the optimality conditions of the least-squares problem for minimizing the measurement residuals due to orientation errors. These optimality constraints form a system of polynomial equations, whose solutions (i.e., all *critical* points of the least-squares cost function) are efficiently computed using eigendecomposition of a so-called multiplication matrix. The globally optimal estimates for the sensor's orientation are then the critical points which minimize the least-squares cost function. In the second stage, we compute the sensor's position using ordinary least squares.

Our proposed approach has several advantages compared to existing algorithms: (i) the optimality of the sensor's orientation estimate is guaranteed in a least-squares sense, (ii) no initialization is required, and (iii) the computational complexity of our approach is *linear* in the number of measurements, whereas state-of-the-art algebraic methods have *quadratic* computational complexity [3]. Furthermore, the developed algorithm can be applied without any modification to solve another robot localization problem, namely estimating pose from line-to-plane correspondences [7], [8]. This is particularly useful when a 2D laser scanner is used to localize a robot inside an *a priori* known building. In this case the straight-line segments in the laser scan that correspond to the structural planes of the building are employed to determine the pose of the laser scanner. The details of this application are omitted due to space considerations and the interested reader is referred to [9].

The remainder of this paper is organized as follows. Section II provides an overview of the related literature. Section III presents the least-squares formulation and polynomial optimality conditions for estimating the sensor's orientation, while Section IV describes the employed polynomial solver. In Section V, a least-squares algorithm is presented to estimate the sensor's position given the orientation estimates. The proposed method is validated with extensive simulations and experiments in Section VI. Finally, conclusions and future research directions are provided in Section VII.

II. Related Work

Exploiting line correspondences to estimate camera pose has received significant attention in the last two decades. In one of the earliest works, Liu *et al.* [1] propose a method using eight or more measurements to linearly constrain the elements of a rotation matrix expressing the sensor's orientation with respect to the global frame. An up-to-scale estimate of the rotation matrix is then obtained by

The authors are with the Dept. of Computer Science and Engineering, University of Minnesota, USA {faraz|stergios}@cs.umn.edu

This work was supported by the University of Minnesota (DTC), and the National Science Foundation (IIS-0643680, IIS-0811946, IIS-0835637). F. M. Mirzaei was supported in part by the UMN Doctoral Dissertation Fellowship.

(linear) least squares, followed by a constraint to ensure the Frobenius norm of the estimated matrix is three. This approach, however, does not necessarily result in a proper orthonormal matrix for noisy measurements.

In [2], Chen proposes an algebraic method to find pose from line correspondences using only three measurements (i.e., the minimum number of measurements required). He also investigates the necessary conditions under which the problem has a finite number of solutions. While the recovered pose is precise in the noise-free case, it is highly unstable in the presence of noise (partially due to utilization of minimal number of measurements), and often produces complex solutions [2]. In addition, this method cannot exploit more than three measurements.

The state-of-the-art algebraic approach to estimate pose from line correspondences, presented by Ansar and Daniilidis in [3], employs *lifting* to convert the polynomials describing four or more measurement constraints to linear equations in the components of the rotation matrix. While this method recovers the orientation precisely in the absence of noise, its performance degrades with increasing measurement-noise variance, and it may even result in complex solutions. Additionally, the lifting method is only guaranteed to work if the polynomial system has exactly one solution. Therefore, in singular configurations where an observed image can correspond to multiple different camera orientations (e.g., when the 3D lines are orthogonal to each other [10]), this method may fail. Moreover, this algorithm has $\mathcal{O}(N^2)$ computational complexity in the number, N, of line measurements used, which can be prohibitive when processing resources are limited.

The main drawback of the aforementioned algebraic methods is that they attempt to solve measurement constraints which are only satisfied in the absence of noise. In the presence of noise and disturbances, however, the coefficients of the polynomials describing the measurement constraints are perturbed. The solutions of a perturbed polynomial system though are extremely unreliable approximations of the roots of the unperturbed system [11, Ch. 2]. In particular, in many instances the solutions of the perturbed system become complex numbers whose real parts are arbitrarily far from the roots of the unperturbed system. To address this issue, one must explicitly account for the measurement noise and formulate the problem as nonlinear least squares with the objective to minimize the measurements' residuals.

In the literature, several iterative (linearization-based) methods have been applied to estimate pose from line correspondences based on nonlinear least-squares minimization. Kumar and Hanson [4] present an iterative least-squares algorithm for recovering the sensor pose. In [5], two iterative methods based on para-perspective and weak-perspective camera models are proposed which show better convergence performance compared to a perspective model in the absence of good initialization. David *et al.* [6] propose an iterative method for finding the camera pose with ambiguous data association. All these methods are *iterative*, and since the nonlinear least-squares cost function is nonconvex, they may converge to a local minimum or a saddle point, and cannot make any claims regarding global optimality.[1] Furthermore, in the absence of a good initialization, these methods are typically slow, and often diverge.

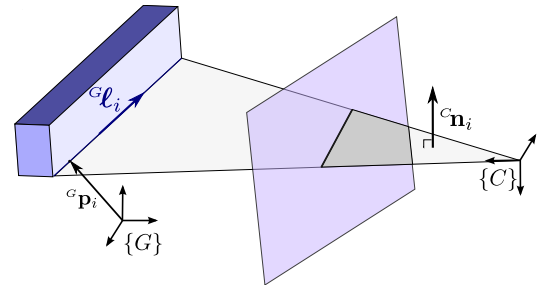


Fig. 1. The i-th 3D line is described in the global frame $\{G\}$ by its direction ${}^G\boldsymbol{\ell}_i$, and its moment ${}^G\mathbf{m}_i = {}^G\mathbf{p}_i \times {}^G\boldsymbol{\ell}_i$, where ${}^G\mathbf{p}_i$ is any arbitrary point on the line. Camera observations of the i-th 3D line can be represented as the projection plane (colored in gray) passing through the 3D line and the optical center of the camera. This plane is described by the normal vector ${}^C\mathbf{n}_i$ expressed in the camera frame. The observed 2D line is the intersection of this plane and the image plane (colored in violet).

In order to address these limitations, in this paper we introduce an algebraic method for solving the nonlinear least-squares pose estimation problem from line correspondences. The main advantages of our approach are: (i) it does not require initialization; (ii) it guarantees the global optimality of the estimated sensor orientation in a least-squares sense; and (iii) it computes all possible poses, if more than one solution exists. Specifically, we formulate the nonlinear least-squares problem for minimizing the measurement residuals due to orientation errors and find *all* its critical points by directly solving the system of polynomial equations describing the optimality (minimization) conditions. This multivariate polynomial system is solved in linear (in the number of measurements) time by efficient construction and eigen-decomposition of the so-called multiplication matrix [11]. Subsequently, the objective function is evaluated at all critical points and the one(s) that results in the smallest cost is selected as the global minimizer. Finally, once the sensor's orientation is determined, we compute its position using linear least squares.

III. Problem Formulation

As mentioned before the most challenging part of pose determination from line correspondences is estimating the camera's attitude since it requires solving a set of nonlinear (polynomial) equations. In the following two sections, we present our method for estimating the camera's attitude, while position determination, given the estimated attitude, is described in Section V.

We assume that N 3D lines with known coordinates (e.g., *a priori* mapped edges of doors and intersections of walls, ceiling, and floor) and their corresponding 2D projections on

[1] Notice that convergence to a local minimum (or generally a stationary point) is due to the nonconvexity of the cost function (e.g., see [12]), and is a completely separate issue from the numerical stability or the convergence properties of these algorithms. Thus this issue *cannot* be addressed by using numerically robust (e.g., conjugate-gradient type) iterative least-squares methods.

the image plane of a pinhole camera are given. Let us define the i-th *projection plane*, $i = 1, \ldots, N$, as the plane that passes through the origin of the camera and the image of the i-th line in that camera (see Fig. 1). The normal vector of this plane can be obtained from the line measured by the camera. Specifically, if we represent the image of the i-th line with polar parameters (θ_i, ρ_i), then every point $[u\ v\ 1]^T$ belonging to that line satisfies the equation $u\cos\theta_i + v\sin\theta_i + \rho_i = 0$, and ${}^C\mathbf{n}_i = [\cos\theta_i\ \ \sin\theta_i\ \ \rho_i]^T$ is the normal vector of the corresponding projection plane.[2] From Fig. 1, it is clear that in the absence of noise, a 3D line lays on its corresponding projection plane. Therefore, if we denote the *a priori* known direction of the i-th 3D line with ${}^G\boldsymbol{\ell}_i$, the unknown orientation of the camera in the global frame, represented by the rotation matrix ${}^C_G\mathbf{C}^T$, satisfies the following constraint:

$$ {}^C\mathbf{n}_i^T\, {}^C_G\mathbf{C}\, {}^G\boldsymbol{\ell}_i = 0. \tag{1} $$

In the presence of noise, we do not have perfect measurements of ${}^C\mathbf{n}_i$. Instead, we measure ${}^C\hat{\mathbf{n}}_i = {}^C\mathbf{n}_i + {}^C\tilde{\mathbf{n}}_i$, where ${}^C\tilde{\mathbf{n}}_i$ is the 3×1 vector of zero-mean Gaussian noise with known covariance $\mathbf{R}_{\mathbf{n},i}$. Substituting ${}^C\mathbf{n}_i$ in (1) yields:

$$ {}^C\hat{\mathbf{n}}_i^T\, {}^C_G\mathbf{C}\, {}^G\boldsymbol{\ell}_i = {}^C\tilde{\mathbf{n}}_i^T\, {}^C_G\mathbf{C}\, {}^G\boldsymbol{\ell}_i \triangleq \eta_i. \tag{2} $$

The measurement *residual*, η_i, is a zero-mean Gaussian random variable with variance $\sigma_i^2 \triangleq {}^G\boldsymbol{\ell}_i^T\, {}^C_G\mathbf{C}^T\, \mathbf{R}_{\mathbf{n},i}\, {}^C_G\mathbf{C}\, {}^G\boldsymbol{\ell}_i$.

Given several noisy line correspondences, the objective is to estimate ${}^C_G\mathbf{C}$. This can be achieved by minimizing the following cost function:

$$ {}^C_G\widehat{\mathbf{C}} = \arg\min_{\mathbf{C}}\ \frac{1}{2}\sum_{i=1}^{N} \sigma_i^{-2}\left({}^C\hat{\mathbf{n}}_i^T\, \mathbf{C}\, {}^G\boldsymbol{\ell}_i\right)^2 \tag{3} $$

$$ \text{subject to } \mathbf{C}^T\mathbf{C} = \mathbf{I}_3,\ \ \det(\mathbf{C}) = 1. \tag{4} $$

This nonlinear weighted least-squares problem for $N \geq 3$ can be solved using iterative methods such as Gauss-Newton [13]. However, iterative approaches often converge to local minima, and require an accurate initial estimate. To address these limitations, we hereafter present a new algebraic method that *directly* solves the nonlinear least-squares problem without requiring initialization.

We start by expressing the orientation of the sensor using the Cayley–Gibbs–Rodriguez (CGR) parametrization since (i) the components of the rotation matrix are naturally expressed as rational functions of the CGR parameters, and (ii) CGR is a minimal representation of rotation, and thus, does not require additional constraints such as the ones in (4) to ensure that it corresponds to a valid rotation [14]. Furthermore, the CGR parametrization introduces the minimum number of unknowns in the resulting polynomial system and hence allows fast computation of its solutions.

In CGR representation, a rotation matrix is expressed as

$$ \mathbf{C} = \left(\mathbf{I}_3 - \lfloor \mathbf{s}\times\rfloor\right)^{-1}\left(\mathbf{I}_3 + \lfloor \mathbf{s}\times\rfloor\right) \tag{5} $$

where $\mathbf{s}^T = [s_1\ \ s_2\ \ s_3]$ is the vector of CGR parameters, and $\lfloor \mathbf{s}\times\rfloor$ is the corresponding skew-symmetric matrix. Equation (5) can be expanded as

$$ \mathbf{C} = \frac{\bar{\mathbf{C}}}{1+\mathbf{s}^T\mathbf{s}}, \quad \bar{\mathbf{C}} \triangleq \left((1-\mathbf{s}^T\mathbf{s})\mathbf{I}_3 + 2\lfloor \mathbf{s}\times\rfloor + 2\mathbf{s}\mathbf{s}^T\right). \tag{6} $$

Substituting this expression in the constraint equation (1), and multiplying both sides by $(1+\mathbf{s}^T\mathbf{s})$ yields:

$$ {}^C\mathbf{n}_i^T\, {}^C_G\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i = 0. \tag{7} $$

This constraint is linear in the components of ${}^C_G\bar{\mathbf{C}}$ which are, in turn, quadratic monomials in the elements of $\mathbf{s}$ [cf. (6)]. In the presence of noise, substituting ${}^C\mathbf{n}_i = {}^C\hat{\mathbf{n}}_i - {}^C\tilde{\mathbf{n}}_i$ in (7) yields

$$ {}^C\hat{\mathbf{n}}_i^T\, {}^C_G\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i = {}^C\tilde{\mathbf{n}}_i^T\, {}^C_G\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i \triangleq \bar{\eta}_i \tag{8} $$

where the measurement residual, $\bar{\eta}_i$, is a zero-mean Gaussian random variable with variance $\bar{\sigma}_i^2 \triangleq {}^G\boldsymbol{\ell}_i^T\, {}^C_G\bar{\mathbf{C}}^T\, \mathbf{R}_{\mathbf{n},i}\, {}^C_G\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i$. Based on (8), we form the following weighted least-squares problem for estimating $\hat{\mathbf{s}}$ from multiple noisy line observations:

$$ \hat{\mathbf{s}} = \arg\min_{\mathbf{s}} J, \qquad J \triangleq \frac{1}{2}\sum_{i=1}^{N} \bar{\sigma}_i^{-2}\left({}^C\hat{\mathbf{n}}_i^T\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i\right)^2. \tag{9} $$

Compared to (3)-(4) the optimization constraint is now removed since the Cayley transformation [cf. (5)] ensures the orthonormality of the estimated rotation matrix. Minimizing J, however, turns out to be computationally intractable, as the degree of the polynomials describing its optimality conditions quickly increases with the addition of each new measurement. For example, given three measurements, the optimality conditions for minimization of (9) will be polynomials of degree 63, while for 10 measurements, they will be of degree $2^{20}-1$.

To mitigate this challenge, we relax the problem by assuming that the variance of the measurement residuals, $\bar{\sigma}_i^2$, is approximately the same for all measurements. This relaxation yields the following least-squares problem:

$$ \hat{\mathbf{s}} = \arg\min_{\mathbf{s}} J', \qquad J' \triangleq \frac{1}{2}\sum_{i=1}^{N} \left({}^C\hat{\mathbf{n}}_i^T\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i\right)^2. \tag{10} $$

To algebraically find the global minimum of (10), we first determine all the critical points of J' by solving the following *optimality conditions* [12]:

$$ f_j(\mathbf{s}) = \frac{\partial J'}{\partial s_j} = \sum_{i=1}^{N}\left({}^C\hat{\mathbf{n}}_i^T\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i\right)\frac{\partial}{\partial s_j}\left({}^C\hat{\mathbf{n}}_i^T\bar{\mathbf{C}}\, {}^G\boldsymbol{\ell}_i\right) = 0 \tag{11} $$

for $j = 1, 2, 3$ and $N \geq 3$. For this purpose, we employ the following proposition:

Proposition 1 (Bézout Theorem [11]): Under mild conditions[3], a system of equations composed of n polynomials of degrees $d_1, d_2, \cdots, d_n$, has $d_1 d_2 \cdots d_n$ distinct solutions. In this case, all the solutions can be obtained using the procedure described in Section IV.

As shown in [2], *at least three* measurements from lines with linearly independent directions are required in order to recover the camera's attitude. Note, however, that the three

[2]Throughout this paper, ${}^A\mathbf{x}$ denotes the expression of a vector $\mathbf{x}$ with respect to frame $\{A\}$, ${}^A_B\mathbf{C}$ is the rotation matrix rotating vectors from frame $\{B\}$ to frame $\{A\}$, and ${}^A\mathbf{p}_B$ is the position of the origin of frame $\{B\}$ with respect to frame $\{A\}$. $\mathbf{I}_n$ is the $n\times n$ identity matrix, and $\mathbf{0}_{m\times n}$ is the $m\times n$ matrix of zeros.

[3]The conditions are (i) the system of polynomial equations has no solution at infinity, and (ii) all solutions are of multiplicity one [11]. These conditions are generally satisfied for the pose-from-line-correspondences problem.

optimality conditions [cf. (11)] are always *cubic* polynomials regardless of the number of measurements; thus, according to Proposition 1, the polynomial system describing the optimality conditions has 27 solutions, each of which is a critical point of J'. The globally optimal solutions of (10) are the critical points that minimize J'.[4]

Note that the computational complexity of solving (11) and finding the global minimum does not increase with the addition of measurements, since the degree and number of polynomials expressing the optimality conditions are fixed. Moreover, computing the contribution of all measurements to the coefficients of the cubic polynomials $f_j,\ j = 1, 2, 3$ increases linearly with the number of measurements.

IV. Polynomial System Solver

Once the optimality conditions (11) are expressed as a system of multivariate polynomial equations, there exist several methods for solving them. Amongst them, numerical methods, such as Newton-Raphson, need initialization and may not find all the solutions. Symbolic reduction methods based on the computation of the system's Gröbner basis are capable of finding all roots without any initialization [11]. However, they can only be used for integer coefficients since their application to floating-point numbers suffers from quick accumulation of round-off errors, which in turn, results in incorrect solutions [11]. Instead, we employ a method developed by Auzinger and Stetter [15] that computes a generalization of the companion matrix to systems of multivariate polynomial equations, namely the *multiplication matrix*, whose eigenvalues are the roots of the associated polynomial system. In particular, this method constructs the multiplication matrix by means of an intermediate so-called *Macaulay* matrix that was originally developed to calculate the resultant of a system of polynomial equations [11].

In the following, we first describe a method to construct the Macaulay matrix, and then compute the multiplication matrix using Schur decomposition of the Macaulay matrix. It is important to note (cf. Section IV-C) that the Macaulay matrix needs to be constructed only once in symbolic form (i.e., treating the coefficients of the polynomials as unknown parameters) and then, in each realization of the problem, we substitute the coefficients obtained from the measurements.

A. Constructing the Macaulay Matrix

We start by introducing the necessary notation and provide a brief overview of algebraic geometry concepts that will be used to compute the solutions of (11). For a detailed discussion of this topic, we refer the reader to [11].

We denote a *monomial* in n variables by $\mathbf{x}^\gamma \triangleq x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n},\ \gamma_i \in \mathbb{Z}_{\geq 0}$, and a polynomial in n variables with complex coefficients by $f = \sum_j c_j \mathbf{x}^{\gamma_j}, c_j \in \mathbb{C}$. The degree of each monomial is defined as $\sum_{i=1}^n \gamma_i$, and the degree of a polynomial is the maximum degree of all its monomials. We assume that the given system of equations has n polynomials, denoted by $f_i = 0,\ i = 1, \ldots, n$, each of them with degree d_i. We define an auxiliary *linear* polynomial, so-called u-polynomial, as $f_0 = u_0 + u_1 x_1 + \cdots + u_n x_n$, where u_i are independently drawn random numbers. Notice that, in general, f_0 will not be zero at the roots of the given system of polynomial equations.

We proceed with defining the total degree of the system of equations, including the auxiliary polynomial as $d \triangleq \sum_{i=0}^n d_i - n = 1 + \sum_{i=1}^n d_i - n$. Then we define the set of all possible monomials of degree less than or equal to d as $\mathcal{S} = \{\mathbf{x}^\gamma \ : \ \sum_j \gamma_j \leq d\}$. It can be easily shown that $\mathcal{S}$ has $\binom{n+d}{n}$ members [11]. For illustration purposes, consider the following system of $n = 2$ polynomials:

$$f_1 = x_1 + 2x_2 + 5 \quad , \quad f_2 = x_1^2 + x_2^2 - 100. \tag{12}$$

In this example, $d_1 = 1$ and $d_2 = 2$, and the total degree after adding the auxiliary polynomial $f_0 = u_0 + u_1 x_1 + u_2 x_2$ is $d = 2$. The set of monomials with degree less than or equal to $d = 2$ is $\mathcal{S} = \{1, x_1, x_2, x_1 x_2, x_1^2, x_2^2\}$.

In the next step, we partition $\mathcal{S}$ into $n+1$ disjoint subsets:

$$\begin{aligned}
\mathcal{S}_n &= \{\mathbf{x}^\gamma \ : \mathbf{x}^\gamma \in \mathcal{S};\ x_n^{d_n} \text{ divides } \mathbf{x}^\gamma\} \\
\mathcal{S}_{n-1} &= \{\mathbf{x}^\gamma \ : \mathbf{x}^\gamma \in \mathcal{S}, \notin \mathcal{S}_n;\ x_{n-1}^{d_{n-1}} \text{ divides } \mathbf{x}^\gamma\} \\
&\vdots \\
\mathcal{S}_0 &= \{\mathbf{x}^\gamma \ : \mathbf{x}^\gamma \in \mathcal{S}, \notin \mathcal{S}_n, \ldots, \notin \mathcal{S}_1\}
\end{aligned}$$

which, for the example system in (12), yields $\mathcal{S}_2 = \{x_2^2\},\ \mathcal{S}_1 = \{x_1^2, x_1 x_2, x_1\},\ \mathcal{S}_0 = \{1, x_2\}$.

Note that in this partitioning $|\mathcal{S}_0| = d_1 d_2 \cdots d_n$, where $|\cdot|$ denotes the cardinality of a set. This is easy to see if we consider that since $x_i^{d_i},\ i = 1, \ldots, n$ do not divide $\mathbf{x}^\gamma \in \mathcal{S}_0$, the power γ_j of each factor x_j in $\mathbf{x}^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} \in \mathcal{S}_0$ can be any integer such that $0 \leq \gamma_j < d_j$. Clearly, under this condition there exist $d_1 d_2 \cdots d_n$ possible choices for $\gamma_j,\ j = 1, \ldots, n$, and accordingly, the same number of distinct monomials belonging to $\mathcal{S}_0$. Additionally, observe that when $|\mathcal{S}_0| > n$, the set $\mathcal{S}_0$ contains 1 and all the monomials $x_1, x_2, \ldots, x_n$. Later on, we will use this important fact to retrieve the solutions of the polynomial system from the eigenvectors of the multiplication matrix.

Based on $\mathcal{S}_0, \ldots, \mathcal{S}_n$, we define the following sets of monomials

$$\mathcal{S}_i' = \left\{ \frac{\mathbf{x}^\gamma}{x_i^{d_i}} \ : \mathbf{x}^\gamma \in \mathcal{S}_i \right\}, \quad i = 1, \ldots, n, \quad \mathcal{S}_0' = S_0$$

and generate an extended set of polynomials by multiplying each polynomial f_i by all monomials in the corresponding $\mathcal{S}_i'$:

$$\begin{aligned}
g_{0,j} &\triangleq \mathbf{x}^{\gamma_j} f_0,\ j = 1, \ldots, |\mathcal{S}_0'|, \quad \text{for each } \mathbf{x}^{\gamma_j} \in \mathcal{S}_0' \\
&\vdots \\
g_{n,j} &\triangleq \mathbf{x}^{\gamma_j} f_n,\ j = 1, \ldots, |\mathcal{S}_n'|, \quad \text{for each } \mathbf{x}^{\gamma_j} \in \mathcal{S}_n'.
\end{aligned}$$

Note that by construction, we have $|\mathcal{S}_0' \cup \cdots \cup \mathcal{S}_n'| = |\mathcal{S}| = \binom{n+d}{n}$ extended polynomials. For the example system (12),

[4] In general, when $N > 3$ lines are observed, there exists a unique global minimum.

we have $\mathcal{S}_2' = \{1\},\ \mathcal{S}_1' = \{x_1, x_2, 1\},\ \mathcal{S}_0' = \{1, x_2\}$, and

$$\begin{aligned} g_{0,1} &= u_0 + u_1 x_1 + u_2 x_2 \\ g_{0,2} &= u_0 x_2 + u_1 x_1 x_2 + u_2 x_2^2 \\ g_{1,1} &= x_1^2 + 2x_2 x_1 + 5x_1 \\ g_{1,2} &= x_1 x_2 + 2x_2^2 + 5x_2 \\ g_{1,3} &= x_1 + 2x_2 + 5 \\ g_{2,1} &= x_1^2 + x_2^2 - 100. \end{aligned}$$

Since the members of $\mathcal{S}_i$ have degrees less than or equal to d, the members of $\mathcal{S}_i'$ will have degrees less than or equal to $d - d_i$. Therefore, all the monomials of $g_{i,j}$ are of degree less than or equal to d. This enables us to express them as a linear combination of the elements of $\mathcal{S}$ (recall that by construction, $\mathcal{S}$ contains all the monomials with degree up to d). We write this linear combination as the inner product of a vector of coefficients $\mathbf{c} \triangleq [c_1\ c_2\ \cdots\ c_\ell]^T$, and $\underline{\mathbf{x}}^{\boldsymbol{\gamma}} \triangleq [\mathbf{x}^{\boldsymbol{\gamma}_1}\ \mathbf{x}^{\boldsymbol{\gamma}_2}\ \cdots\ \mathbf{x}^{\boldsymbol{\gamma}_\ell}]^T$ with $\mathbf{x}^{\boldsymbol{\gamma}_i} \in \mathcal{S}$ and $\ell \triangleq |\mathcal{S}| = \binom{n+d}{n}$, i.e.,

$$g_{i,j} = \mathbf{x}^{\boldsymbol{\gamma}_j} f_i = \mathbf{c}_{i,j}^T \underline{\mathbf{x}}^{\boldsymbol{\gamma}},\ i = 0, \ldots, n,\ j = 1, \ldots, |\mathcal{S}_i'|.$$

Stacking together all available $g_{i,j}$ polynomials and arranging $\underline{\mathbf{x}}^{\boldsymbol{\gamma}} = [\underline{\mathbf{x}}^{\boldsymbol{\alpha}}\ \underline{\mathbf{x}}^{\boldsymbol{\beta}}]^T$, where $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$ are monomials of $\mathcal{S}_0$ and $\underline{\mathbf{x}}^{\boldsymbol{\beta}}$ are the rest of the monomials, yields:

$$\begin{bmatrix} g_{0,1}(\underline{\mathbf{x}}^{\boldsymbol{\gamma}}) \\ g_{0,2}(\underline{\mathbf{x}}^{\boldsymbol{\gamma}}) \\ \vdots \\ g_{1,1}(\underline{\mathbf{x}}^{\boldsymbol{\gamma}}) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{0,1}^T \\ \mathbf{c}_{0,2}^T \\ \vdots \\ \mathbf{c}_{1,1}^T \\ \vdots \end{bmatrix} \underline{\mathbf{x}}^{\boldsymbol{\gamma}} = M \underline{\mathbf{x}}^{\boldsymbol{\gamma}} = M \begin{bmatrix} \underline{\mathbf{x}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{x}}^{\boldsymbol{\beta}} \end{bmatrix}. \tag{13}$$

The *Macaulay matrix*, M, is a square matrix of dimension $|\mathcal{S}| = \binom{n+d}{n}$ comprising the coefficients of $f_0, \ldots, f_n$. This matrix produces the extended set of polynomials $g_{i,j}$ from the vector of monomials $\underline{\mathbf{x}}^{\boldsymbol{\gamma}}$, and plays an important role in computing the resultant of a system of polynomial equations [11]. In the next section, we describe the process for extracting the multiplication matrix from the Macaulay matrix and for finding the roots of the polynomial system (11).

B. Computing the Roots of the Polynomial System

Let $\mathbf{p} = [p_1\ \cdots\ p_n]^T$ be a solution of the system of polynomial equations, i.e., $f_1(\mathbf{p}) = \cdots = f_n(\mathbf{p}) = 0$ [cf. (11)], and thus $g_{1,1}(\mathbf{p}) = \cdots = g_{n,|\mathcal{S}_n'|}(\mathbf{p}) = 0$ (note that $f_0(\mathbf{p})$ and $g_{0,j}(\mathbf{p})$ are not generally zero). Denoting the vector of monomials $\underline{\mathbf{x}}^{\boldsymbol{\gamma}}$ evaluated at $\mathbf{p}$ as $\underline{\mathbf{p}}^{\boldsymbol{\gamma}}$, and substituting in (13), yields:

$$\begin{bmatrix} g_{0,1}(\mathbf{p}) \\ \vdots \\ g_{0,|\mathcal{S}_0|}(\mathbf{p}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = M \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \Leftrightarrow \begin{bmatrix} f_0(\mathbf{p})\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = M \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{14}$$

where by construction, $[g_{0,1}\ \cdots\ g_{0,|\mathcal{S}_0|}]^T = f_0\, \underline{\mathbf{x}}^{\boldsymbol{\alpha}}$. We introduce the partitioning $M = \left[\begin{smallmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{smallmatrix}\right]$ where M_{00} is of dimensions $|\mathcal{S}_0| \times |\mathcal{S}_0|$, and the other submatrices are of compatible size, and write (14) as

$$\begin{bmatrix} f_0(\mathbf{p})\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{15}$$

Employing the Schur complement of M, we obtain

$$f_0(\mathbf{p})\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} = \widetilde{M}\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \tag{16}$$

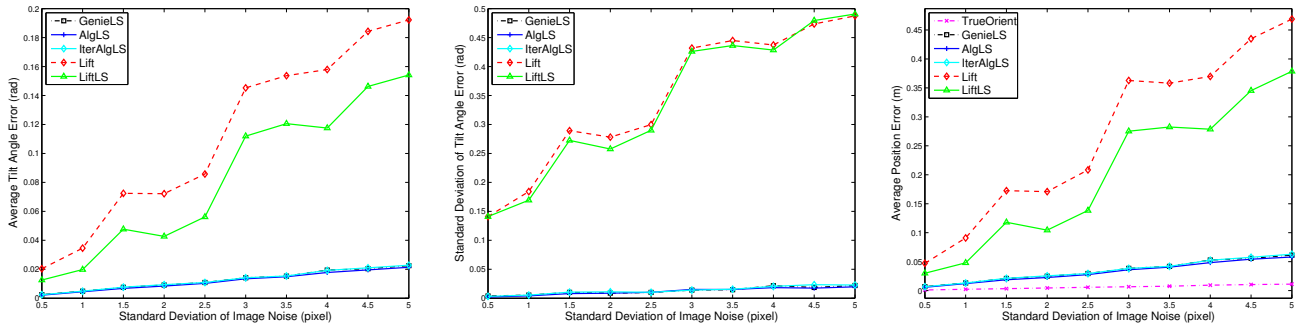
where $\widetilde{M} = M_{00} - M_{01} M_{11}^{-1} M_{10}$ has dimensions $|\mathcal{S}_0| \times |\mathcal{S}_0|$ and is the *multiplication matrix*.

As evident from (16), the vector of monomials of $\mathcal{S}_0$ (i.e., $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$), evaluated at a root $\mathbf{p}$ of the original polynomial system, is an eigenvector of $\widetilde{M}$, while $f_0(\mathbf{p})$ is the corresponding eigenvalue. Therefore, to find the 27 solutions of our polynomial system [cf. (11)], we first compute the eigenvectors of $\widetilde{M}$ which has dimension 27. Then, considering that one of the monomials in $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$ should be equal to 1, we scale the eigenvectors such that their components corresponding to this monomial become one. Finally, the roots of (11) appear in the elements of the scaled eigenvectors that correspond to the monomials $x_1, x_2, \ldots, x_n$, $n = 3$ in the vector $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$ (note that for the case of (11) $|\mathcal{S}_0| = 27 > n = 3$).

C. Implementation Remarks

The method described for determining the roots of (11) can be implemented very efficiently. Note that the construction of the Macaulay matrix is independent of the explicit values of each polynomial's coefficients since the degrees of the polynomials remain the same. Hence, although the coefficients change in each realization of the problem (as they depend on the measurements), we can treat each coefficient as a symbolic parameter, and construct the Macaulay matrix (of dimension $|\mathcal{S}| = 120$) as a function of these parameters off-line (e.g., using publicly available Maple packages [16], [17]). For each realization of the problem, we (i) replace the symbolic parameters of the Macaulay matrix with floating-point coefficients obtained from the measurements, (ii) compute its Schur complement to obtain the multiplication matrix $\widetilde{M}$ (of dimension $|\mathcal{S}_0| = 27$), (iii) determine the eigenvectors of $\widetilde{M}$, and (iv) read the roots of (11) from the eigenvectors' corresponding elements after scaling. Note that the 27 roots of (11) are the critical points of (10). To find the global minimum, we substitute them in (10) and select the one(s) that minimizes the cost function J'.

In practice, M_{11} in (15) may be bad conditioned or even rank deficient, preventing accurate computation of the Schur complement of M. This can happen in several situations: (i) when the u-polynomial f_0 is close to zero at the solution of the polynomial system; (ii) when the rotation angle corresponding to ${}^C_G\bar{\mathbf{C}}$ is close to $180°$ (leading to extremely large CGR parameters); (iii) if the 3D line directions ${}^G\boldsymbol{\ell}_i$ or the line measurements ${}^C\mathbf{n}_i$ have one or two zero components (e.g., when 3D lines are aligned with the cardinal axes). The first problem is easily addressed by re-generating another random u-polynomial. The last two problems are resolved by rotating the measurements or 3D lines to an arbitrary (randomly generated) frame of reference, finding the global minimum(s), and then rotating the solutions back to the original frame.



(a) (b) (c)

Fig. 2. Monte Carlo simulation results for different standard deviations of the image noise when 5 lines are observed: (a) Average tilt-angle error; (b) Standard deviation of the tilt-angle error; and (c) Average position error.

V. Estimation of Sensor Position

Once the orientation of the camera is known, we can easily compute its position, ${}^G\mathbf{p}_C$ using the *a priori* known *moments* of the 3D lines, expressed in the global frame as ${}^G\mathbf{m}_i \triangleq {}^G\mathbf{p}_i \times {}^G\boldsymbol{\ell}_i, i \geq 3$, where ${}^G\mathbf{p}_i$ is any point on the 3D line (see Fig. 1). Following the same convention, the moment of the i-th line expressed in the camera frame is ${}^C\mathbf{m}_i = {}^C\mathbf{p}_i \times {}^C\boldsymbol{\ell}_i$, where ${}^C\boldsymbol{\ell}_i = {}^C_G\mathbf{C}\,{}^G\boldsymbol{\ell}_i$, and ${}^C\mathbf{p}_i = {}^C_G\mathbf{C}\,({}^G\mathbf{p}_i - {}^G\mathbf{p}_C)$. Expanding ${}^C\mathbf{m}_i$ yields:

$$\begin{aligned} {}^C\mathbf{m}_i &= {}^C\mathbf{p}_i \times {}^C\boldsymbol{\ell}_i \\ &= {}^C_G\mathbf{C}\,({}^G\mathbf{p}_i - {}^G\mathbf{p}_C) \times {}^C_G\mathbf{C}\,{}^G\boldsymbol{\ell}_i \\ &= {}^C_G\mathbf{C}\,{}^G\mathbf{m}_i + {}^C_G\mathbf{C}\,\lfloor {}^G\boldsymbol{\ell}_i \times \rfloor {}^G\mathbf{p}_C. \end{aligned} \tag{17}$$

Although ${}^C\mathbf{m}_i$ cannot be measured directly, one can easily check that it is perpendicular to the projection plane of the i-th line. Therefore, in the absence of noise, any point ${}^C\mathbf{w}_i = [u_i \;\; v_i \;\; 1]^T$ that lays on the image of the i-th line, and thus the projection plane, satisfies the constraint:

$${}^C\mathbf{w}_i^T\,{}^C\mathbf{m}_i = 0. \tag{18}$$

In particular, if the image of the i-th line is parametrized as $u\cos\theta_i + v\sin\theta_i + \rho_i = 0$ (with θ_i and ρ_i computed using least-squares line fitting [18]), then we choose ${}^C\mathbf{w}_i = [-\rho_i\cos\theta_i \;\; -\rho_i\sin\theta_i \;\; 1]^T$. Substituting (17) in (18) we obtain:

$${}^C\mathbf{w}_i^T({}^C_G\widehat{\mathbf{C}}\,{}^G\mathbf{m}_i + {}^C_G\widehat{\mathbf{C}}\,\lfloor {}^G\boldsymbol{\ell}_i \times \rfloor {}^G\mathbf{p}_C) = 0 \tag{19}$$

where we have replaced ${}^C_G\mathbf{C}$ with its estimate ${}^C_G\widehat{\mathbf{C}}$ (cf. (5) and Section IV). Given measurements to at least three lines with linearly independent directions [2], the following system of equations can be solved using ordinary least squares to obtain an estimate for the camera's position in the global frame of reference:

$$\begin{bmatrix} {}^C\mathbf{w}_1^T\,{}^C_G\widehat{\mathbf{C}}\,\lfloor {}^G\boldsymbol{\ell}_1 \times \rfloor \\ {}^C\mathbf{w}_2^T\,{}^C_G\widehat{\mathbf{C}}\,\lfloor {}^G\boldsymbol{\ell}_2 \times \rfloor \\ \vdots \end{bmatrix} {}^G\hat{\mathbf{p}}_C = \begin{bmatrix} -{}^C\mathbf{w}_1^T\,{}^C_G\widehat{\mathbf{C}}\,{}^G\mathbf{m}_1 \\ -{}^C\mathbf{w}_2^T\,{}^C_G\widehat{\mathbf{C}}\,{}^G\mathbf{m}_2 \\ \vdots \end{bmatrix}. \tag{20}$$

VI. Simulation and Experimental Results

A. Simulations

We hereafter present Monte Carlo simulation results that confirm the superior performance of our proposed pose-from-line-correspondences algorithm over existing approaches. Specifically, we compare the error in the estimated camera attitude obtained from each of the following algorithms:

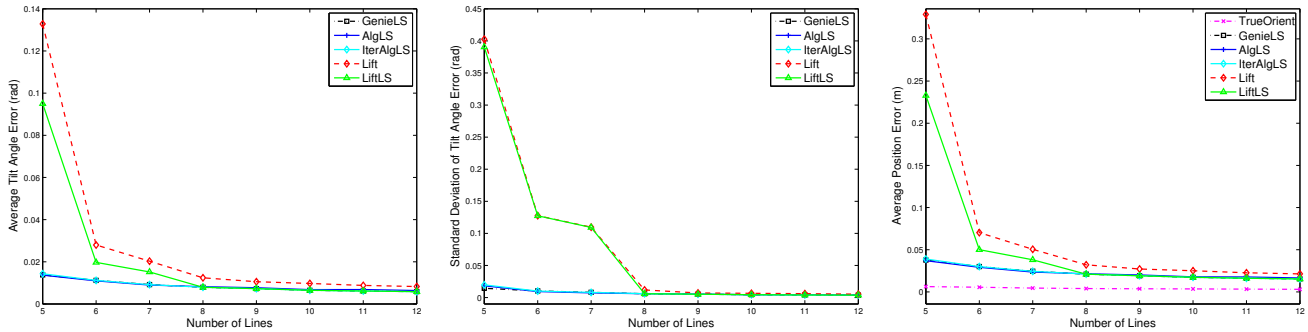
- **Lift:** Lifting method of Ansar and Daniilidis [3].
- **LiftLS:** Weighted least squares proposed in [4], and initialized using the estimates from **Lift**.
- **AlgLS:** Our proposed single-step algebraic minimization of the relaxed least-squares cost function [cf. (10)].
- **GenieLS:** Weighted least-squares minimization of (3) initialized with the true orientation (used as a benchmark).
- **IterAlgLS:** Iterative algebraic minimization of the weighted least squares cost function [cf. (9)].

Note that **IterAlgLS** is our proposed extension to **AlgLS** for iteratively solving (9). The first iteration of **IterAlgLS** is the same as **AlgLS**. In the following iterations, the estimated orientation is used to approximate $\bar{\sigma}_i^2$, whose inverse appears in (9). This allows us to algebraically minimize the original cost function in (9) in a similar way to **AlgLS**.

We evaluate the performance of each algorithm by comparing the estimated and real pose. Specifically, we use the *norm of the tilt angle error* for assessing orientation accuracy. Let us denote by $\mathbf{C}$ and $\widehat{\mathbf{C}}$ the true and estimated rotation matrices, respectively. Then the orientation error is $\widetilde{\mathbf{C}} = \mathbf{C}^T\widehat{\mathbf{C}}$. After we convert $\widetilde{\mathbf{C}}$ to CGR parametrization and denote it as $\tilde{\mathbf{s}}$, then the tilt angle error can be obtained using the small-angle approximation as $||\boldsymbol{\delta\theta}|| = 2||\tilde{\mathbf{s}}||$. The error in the estimated position is $||\mathbf{p} - \hat{\mathbf{p}}||$.

The simulation setup is as follows: At each trial, the simulated pinhole camera with focal length of 512 pixels is placed at a random position and orientation with respect to the world. The camera measures pixelated projections of N randomly generated $3D$ line segments of different lengths, perturbed with i.i.d. random Gaussian noise with standard deviation of σ_p pixels. A least-squares line fitting is then employed to find the $2D$ line parameters ρ_i, θ_i from the pixelated line measurements.

We present two sets of simulation results. The first set demonstrates the performance of the aforementioned methods for different standard deviations of the image noise, σ_p, while fixing the number of observed lines to $N = 5$ (Fig. 2). The second set of results evaluates performance when varying the number of lines, N, while fixing the standard deviation of the image noise at $\sigma_p = 3$ pixels. The



(a) (b) (c)

Fig. 3. Monte Carlo simulation results for different numbers of detected lines when the standard deviation of the image noise is 3 pixels: (a) Average tilt-angle error; (b) Standard deviation of the tilt-angle error; (c) Average position error.

results correspond to 1000 trials for each value of N and σ_p (Fig. 3). In all simulations, when multiple global minimizers are obtained, we choose the one closest to the true camera pose. This is reasonable, since in practical situations we can often discard all but one minimizer by considering visibility constraints and re-projection errors.

Results from both simulations confirm the superior performance of our method compared to **Lift** and **LiftLS**. Specifically, **AlgLS** is almost always as good as the weighted least squares initialized with the true orientation (**GenieLS**). As it is expected, **Lift** is the worst in terms of accuracy since it does not account for noise. The least-squares algorithm initialized with the solution of lifting, labeled as **LiftLS**, has better accuracy compared to **Lift**; however, its performance is significantly inferior to our proposed method since it can diverge if its initialization is inaccurate. In Figs. 2(a) and 2(b) it can be seen that the performance of **Lift** and **LiftLS** quickly degrade as the image noise increases. However, **AlgLS** demonstrates significantly better robustness to noise. Similarly, Figs. 3(a) and 3(b) demonstrate that **Lift** and **LiftLS** perform very poorly as the number of line measurements approaches the minimum required.[5]

Figs. 2(c) and 3(c) demonstrate the impact of estimating pose form line correspondences in two steps (i.e., first attitude and then position) on the performance of position estimation. In particular, observe that the error in the estimated position using orientation from **AlgLS** is significantly lower than **Lift**, and very close to the benchmark performance **TrueOrient** where the true orientation is used in (20). Finally, we point out that as evident from Figs. 2 and 3, the results obtained by **IterAlgLS** do not yield significant performance improvement compared to **AlgLS**.

B. Experiments

In order to validate the proposed algorithms in real situations, we have conducted a number of experiments. Specifically, we have taken an image of an object [wooden cube, see Fig. 4(a)] and of an indoor scene [corner of a room, see Fig. 4(e)] of known dimensions using an intrinsically-calibrated Dragonfly Express camera. We have then manually selected several lines in each image corresponding to the 3D lines with known coordinates [see Figs. 4(a) and 4(e)]. We have employed these line correspondences to estimate the camera orientation according to the procedure outlined in Section IV. In both experiments, four global minimizers are found for (10) and for each of them, the corresponding camera position is estimated (cf. Section V). Among these candidate poses, the one that results in the scene to be in front of the camera is selected. Using this selected pose, the known 3D lines (including the ones that were not detected before, for example due to invisibility) are back-projected onto the image [see Figs. 4(b) and 4(f)] to validate the obtained results. The estimated camera poses from **Lift** and **LiftLS** are also used to plot the (back-)projection of the known 3D lines in Figs. 4(c), 4(g), and 4(d), 4(h), respectively. It can be clearly seen that in this experiment both **Lift** and **LiftLS** result in completely wrong camera poses, possibly due to the existence of multiple solutions for the camera orientation. Table I provides a summary of the estimated orientations (vector $\mathbf{s}$) by different methods along with their average execution times (for the corner and the cube experiments) from Matlab implementations on a 2 GHz Core 2 Duo processor. Note that the execution time of **LiftLS** includes initialization by **Lift**.

TABLE I

COMPUTED ORIENTATION, EXPRESSED AS CGR PARAMETERS, AND AVERAGE EXECUTION TIMES FOR "CUBE" AND "CORNER" EXPERIMENTS USING DIFFERENT METHODS.

	cube (using 8 lines)	corner (using 10 lines)	exec. time
AlgLS	$[1.37 \quad 3.96 \quad -2.51]$	$[0.91 \quad 1.26 \quad -1.34]$	25 msec.
Lift	$[-6.99 \quad 3.92 \quad -0.79]$	$[-2.00 \quad 11.94 \quad 27.60]$	190 msec.
LiftLS	$[1.58 \quad -0.55 \quad 0.40]$	$[-1.10 \quad 1.47 \quad 1.38]$	194 msec.

VII. CONCLUSION

In this paper, we have presented an efficient algorithm for precisely estimating a camera's pose given observations of three or more known 3D lines. Contrary to previous algebraic approaches that solve a deterministic (noise-free) version of this problem, our formulation explicitly accounts for the presence of noise in the image measurements. Moreover, in

[5]Note that it is also possible to use the lifting algorithm of [3] with four lines. However, that would require a different implementation than when five lines or more are available, and hence it is not considered here. Note also that according to the simulation results in [3] the lifting algorithm's accuracy for four lines is considerably inferior to that of five or more lines.

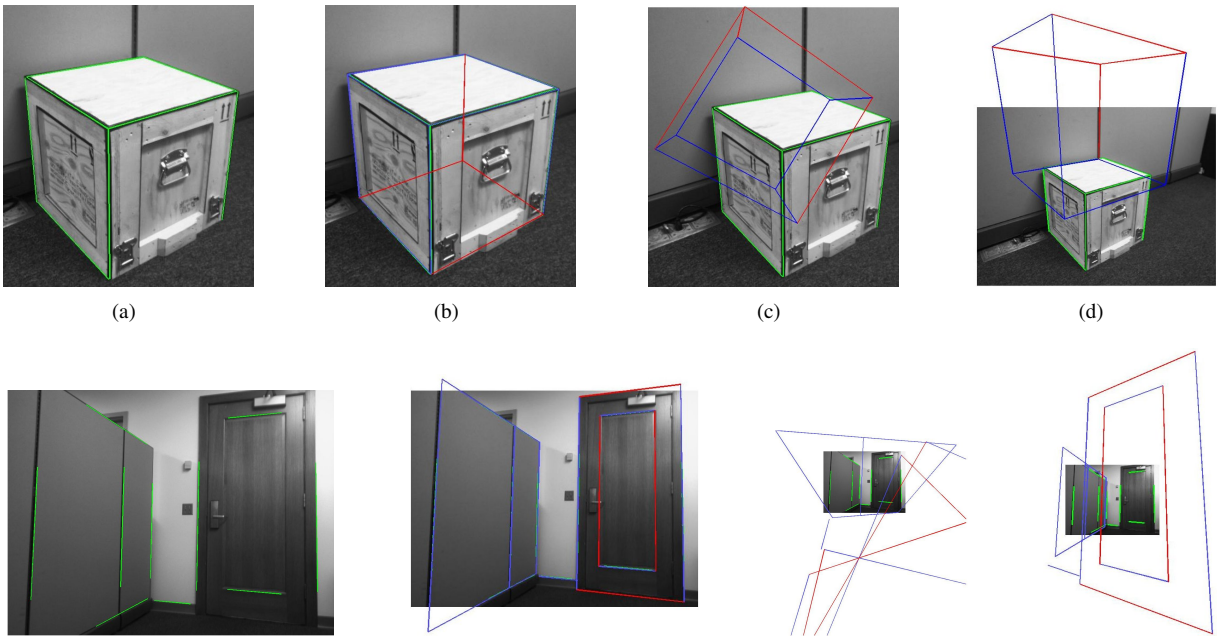


Fig. 4. Camera pose determination with respect to a box, and a corner inside an office, both of known dimensions. Manually selected lines are specified by green color and their back-projections are marked by blue. Projection of invisible (e.g., rear edges of the cube) or previously undetected (e.g., bottom edge of the door) lines are colored as red. (a,e): Initial selection of lines; (Back-) projection of the lines using the estimated camera pose from (b,f): **AlgLS**; (c,g): **Lift**; (d,h): **LiftLS**.

contrast to existing nonlinear least-squares methods, which consider noisy observations but only guarantee convergence to a stationary point (through iterative minimization), our algorithm requires no initial estimate and is guaranteed to find the global minimum of the least-squares cost function for the orientation error. The key idea behind our approach is that the optimality conditions of the nonlinear least-squares problem form a system of multivariate polynomial equations which is directly solved, using algebraic geometry techniques, to determine all the critical points of the cost function, and thus the estimate (global minimum) that minimizes the orientation error. Once the camera's attitude is computed, its position is then determined using ordinary (linear) least squares. Extensive simulation and experimental results demonstrate that our algorithm significantly outperforms existing methods and achieves accuracy almost indistinguishable from that of an (ideal) iterative least-squares estimator initialized with the true camera orientation.

Currently, we are investigating efficient methods for data association within the same framework and also extensions that consider the presence of known points in the scene.

References

[1] Y. Liu, T. S. Huang, and O. D. Faugeras, "Determination of camera location from 2-d to 3-d line and point correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 28–37, Jan. 1990.

[2] H. H. Chen, "Pose determination from line-to-plane correspondences: existence condition and closed-form solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 530–541, Jun. 1991.

[3] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, May 2003.

[4] R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vis. Graph. Image Process.*, vol. 60, no. 3, pp. 313–342, Nov. 1994.

[5] F. Dornaika and C. Garcia, "Pose estimation using point and line correspondences," *Real-Time Imaging*, vol. 5, no. 3, pp. 215–230, Jun. 1999.

[6] P. David, D. DeMenthon, R. Duraiswami, and H. Samet, "Simultaneous pose and correspondence determination using line features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Madison, WI, Jun. 18–20, 2003, pp. 424–431.

[7] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, "Orthogonal SLAM: a step toward lightweight indoor autonomous navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 9–15, 2006, pp. 5007–5012.

[8] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis, "A laser-aided inertial navigation system (L-INS) for human localization in unknown indoor environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 3–8, 2010, pp. 5376–5382.

[9] F. M. Mirzaei and S. I. Roumeliotis, "Globally optimal pose estimation from line-to-plane correspondences," University of Minnesota, Department of Computer Science, Tech. Rep., 2010.

[10] N. Navab and O. Faugeras, "Monocular pose determination from lines: critical sets and maximum number of solutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York City, NY, Jun. 1993, pp. 254 –260.

[11] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*. Springer, 2004.

[12] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 2004.

[13] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River: Prentice Hall, 1993.

[14] M. D. Shuster, "A survey of attitude representations," *J. Astronaut. Sci.*, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.

[15] W. Auzinger and H. J. Stetter, "An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations," in *Proc. Int. Conf. on Numer. Math.*, Singapore, 1988, pp. 11–30.

[16] M. Minimair, "MR: Macaulay resultant package for Maple," 2005. [Online]. Available: http://minimair.org/mr

[17] L. Busé, I. Emiris, B. Mourrain, O. Ruatta, and P. Trébuchet, "Multires, a Maple package for multivariate resolution problems," 2002. [Online]. Available: http://www-sop.inria.fr/galaad/logiciels/multires

[18] S. Yi, R. M. Haralick, and L. G. Shapiro, "Error propagation in machine vision," *Mach. Vision Appl.*, vol. 7, no. 2, pp. 93–114, Apr. 1994.