Robust camera pose estimation from unknown or known line correspondences

Xiaohu Zhang,[1,2,†] Zheng Zhang,[3,†] You Li,[1,2] Xianwei Zhu,[1,2] Qifeng Yu,[1,2] and Jianliang Ou[1,*]

[1]College of Aerospace and Materials Engineering, National University of Defense Technology, Changsha 410073, Hunan, China

[2]Hunan Key Laboratory of Videometrics and Vision Navigation, Changsha 410073, Hunan, China

[3]School of Computer Engineering, Nanyang Technological University, Singapore

*Corresponding author: videometric@yahoo.cn

[†]The first two authors contributed equally to this work.

Received 17 October 2011; revised 7 December 2011; accepted 17 December 2011; posted 20 December 2011 (Doc. ID 156622); published 28 February 2012

We address the model-to-image registration problem with line features in the following two ways. (a) We present a robust solution to simultaneously recover the camera pose and the three-dimensional-to-two-dimensional line correspondences. With weak pose priors, our approach progressively verifies the pose guesses with a Kalman filter by using a subset of recursively found match hypotheses. Experiments show our method is robust to occlusions and clutter. (b) We propose a new line feature based pose estimation algorithm, which iteratively optimizes the objective function in the object space. Experiments show that the algorithm has strong robustness to noise and outliers and that it can attain very accurate results efficiently.

OCIS codes: 150.1135, 100.2960, 000.4430.

1. Introduction

Model-to-image registration is a basic problem in computer vision and has found many applications in visual navigation, object recognition, augmented reality, and photogrammetry. Its task is to determine the orientation and position of a calibrated camera from a two-dimensional (2D) image of a three-dimensional (3D) object. On one hand, without knowing the 3D-to-2D feature correspondences between the 3D scene object and the 2D image, one needs to simultaneously find the correct correspondences and the camera pose [1–4], which is desirable for automatic registration of 3D models to images or for the situation where it is difficult to establish the correspondences. On the other hand, with known 3D-to-2D feature correspondences, the task is well known as pose estimation [5–12], which is to recover the rigid transformation between the object and the camera coordinate system. These two aspects of model-to-image registration are closely connected, since the simultaneous estimation of both pose and correspondence usually needs to be refined by a pose estimation method. This paper first presents a solution to simultaneously recover the camera pose and the 3D-to-2D line correspondences and then introduces a new line feature based pose estimation algorithm.

Point features and line features are two main features of registration. Although the point feature is often used, the line feature, which has the advantages of robust detection and more structural information, is gaining increasing attention. Typically, in indoor environments, many man-made objects have planar surfaces with uniform color or poor texture, where few point features can be localized, but such objects are abundant in line features that can be localized more stably and accurately. Moreover, line

1559-128X/12/070936-13$15.00/0

features are less likely to be affected by occlusions, thanks to multipixel support. We build our work of model-to-image registration on line features.

Without using any additional information, solving the coupled problem may be easily affected by noise and clutter, leading to rough estimations or even wrong solutions. In actual applications, prior knowledge about the camera pose is usually available and can be used for the pose and correspondence estimation problem [4]. By exploiting weak pose priors, we present a method to simultaneously find the correspondences and estimate the camera pose. Starting from a set of initial pose guesses with an associated uncertainty, our approach progressively estimates the optimal pose and correspondences using a recursive search framework. For each pose guess, a subset of line matches is hypothesized with a Kalman filter update scheme to progressively find the optimal pose. The line projection uncertainty is exploited to significantly reduce the number of 2D image lines that can be potentially matched to a 3D line. The validity and effectiveness of our approach are proven by both synthetic and real data experiments.

With known correspondences of line features, e.g., either given in advance or having been established via the simultaneous pose and correspondence method, a pose estimation algorithm is often needed to estimate or refine the pose. We present an accurate and efficient solution for pose estimation from line features. By introducing coplanarity errors, we formulate the objective functions in terms of distances in the 3D scene space and use different optimization strategies to find the best rotation and translation. We show by experiments that the algorithm that fully exploits the line constraint information can attain accurate and robust results efficiently even under strong noise and outliers.

The rest of this paper is organized as follows. The related work is given in Section 2, followed by the description of the camera model in Section 3. In Section 4, we introduce our simultaneous pose and correspondence approach. In Section 5, we describe our pose estimation method. Experiments are given in Section 6, followed by the conclusion in Section 7.

2. Related Work

We review briefly the work of model-to-image registration in the two aspects here.

A. Simultaneous Pose and Correspondence

Simultaneous pose and correspondence is a coupled problem in that one part becomes trivial once the other has been solved. A typical approach is the hypothesize-and-test method [13]: one first hypothesizes a small set of correspondences and uses them to compute a pose estimate, which is tested as to whether it decreases the backprojection errors to a sufficiently low level, then the pose is accepted; otherwise, this process is repeated. RANSAC [1] is one of the best known such hypothesize-and-test approaches. Many algorithms were proposed since then to improve the hypothesize-and-test method: to reduce the computational burden [13–15], to solve the 3D full perspective matching problem [2,16], and to make use of line features [17,18]. Among them, the SoftPOSIT algorithm [2] proposed by David *et al.* is regarded as the most computationally effective, with a complexity of $\mathbb{O}(MN^2)$ (M and N are the number of 3D and 2D points, respectively). This method has also been extended for matching 3D model lines to 2D image lines [3] but cannot be used for the 3D full perspective situation.

Moreno-Noguer *et al.* [4] solve the simultaneous pose and correspondence problem by introducing weak pose priors to guide the search process. Their method models the pose priors as a Gaussian mixture model (GMM) that is progressively refined by hypothesizing new correspondences. This rapidly reduces the number of potential matches for each 3D point and explores the pose space more thoroughly than SoftPOSIT at a similar computational cost. Inspired by this point feature based method, we present an approach for solving the pose and correspondence problem from line features.

B. Pose Estimation

The problem of pose estimation has been studied for a long time in the community of computer vision and photogrammetry, and numerical methods have been proposed. Most existing approaches solve the problem using point features. In this case, the problem is also known as the perspective-n-point (PnP) problem [12,19–21].

Besides the point feature, the line feature has also received much attention. Closed-form algorithms were derived for three-line correspondences, but multiple solutions may appear [5,22]. A linear solution [10] was proposed for solving the pose estimation problem from n points or n lines. It guarantees a solution for $n > 4$ if the world objects do not lie in a critical configuration. For fast or real-time applications, such closed-form or linear algorithms free of initialization [5,10,22,23] can be used. In order to obtain more accurate results, iterative algorithms based on nonlinear optimization [8,24,25] are generally required. However, they generally do not fully exploit the specific structure of the pose estimation problem, and the usual use of Euler angle parameterization of rotation cannot always enforce the orthogonality constraint of the rotation matrix. Moreover, the typical iterative framework that uses classical optimization techniques such as the Newton and Levenberg–Marquardt methods may lack sufficient efficiency [9,26,27]. One interesting exception among the iterative algorithms is the orthogonal iteration (OI) algorithm developed for point features [9], which is not only accurate but also robust to corrupted data and fast enough for real-time applications. The OI algorithm formulates the pose estimation problem as minimizing an error metric based on collinearity in object space and iteratively computing orthogonal

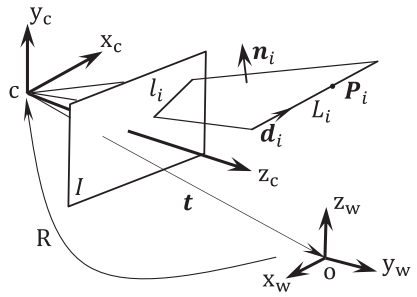


Fig. 1. Geometry of the camera model.

rotation matrices in a globally convergent manner. Inspired by this method, we present a pose estimation algorithm based on line features.

3. Camera Model

Figure 1 illustrates the geometry of the camera model. Let $c - x_c y_c z_c$ be the camera coordinate system with the origin fixed at the optical center point and the axis z_c coinciding with the optical axis and pointing to the front of the camera. I denotes the image plane. We assume the optical axis is perpendicular to the image plane. $o - x_w y_w z_w$ is the object coordinate system. L_i is a 3D line in the space, and l_i is its 2D image projection. It can be seen that the optical center, the 2D image line l_i, and the 3D line L_i are on the same plane, which is called the interpretation plane [5]. In the object coordinate system, L_i can be described as $\lambda \mathbf{d}_i + \mathbf{P}_i$, where $\mathbf{d}_i$ is the unit direction of the line and, $\mathbf{P}_i$ is an arbitrary point on the line, and λ is a scalar. Let us assume the image plane I is the one with the z_c coordinate as 1, i.e., the normalized image plane; the 2D image line l_i in the camera coordinate system can then be expressed as $a_i x + b_i y + c_i = 0$, where $\mathbf{n}_i = (a_i, b_i, c_i)^T$ is defined as a unit vector and $(x, y, 1)^T$ denotes the image point coordinates. It is clear that $\mathbf{n}_i$ is the normal vector of the interpretation plane. Considering line detection in an actual image plane for which the z_c coordinate is decided by the camera intrinsic parameters, the 2D image line is often represented, for example, by $k_i u + v + b_i = 0$ in the pixel coordinate system. Assuming one image point that has pixel coordinates (u, v) has the coordinates (x, y) in the normalized image plane, the transformation between the two coordinate systems is $(u, v, 1)^T = A \cdot (x, y, 1)^T$ (A is the camera intrinsic matrix). From this, we can easily obtain

$$\mathbf{n}_i^T = \mathbf{m}_i^T A, \tag{1}$$

where $\mathbf{m}_i = (k_i, 1, b_i)^T$.

The direction vector $\mathbf{d}_i$ and the point $\mathbf{P}_i$ can be expressed in the camera coordinate system as $R\mathbf{d}_i$ and $R\mathbf{P}_i + \mathbf{t}$, respectively, where the 3×3 rotation matrix R and the translation vector $\mathbf{t}$ describe the rigid transformation between the object coordinate system and the camera coordinate system. Since the two vectors are all in the interpretation plane, we have

$$\mathbf{n}_i^T R \mathbf{d}_i = 0, \tag{2}$$

$$\mathbf{n}_i^T (R\mathbf{P}_i + \mathbf{t}) = 0. \tag{3}$$

Therefore, each line correspondence provides two constraints. Equations (2) and (3) are the well known fundamental equations of the pose estimation problem from 2D-to-3D line correspondences [28].

4. Pose from Unknown Line Correspondences

Simultaneously estimating the camera pose and correspondences is suited for the situation when it is difficult to establish the object-to-image correspondences, for example, because the object contains many repetitive patterns or because it is difficult to extract distinguished image line features. Our algorithm for the simultaneous pose and correspondence problem from lines builds on the Blind PnP algorithm [4], whose key idea is the use of pose priors to guide the search for potential matches and to progressively update the pose hypotheses. A set of available pose priors are represented as a GMM, which includes G Gaussian components. Each Gaussian component, with a six-vector $\mathbf{p}_g$ of mean and a 6×6 matrix Σ_g^p of covariance, provides an initial pose with an associated covariance. With a prior GMM of G components, M 3D object points, and N image points (their correspondences are unknown), the complexity of the Blind PnP algorithm is in the order of $\mathbb{O}(Gn^3M \log N)$, where n is the number of 2D correspondences that can be potentially matched to each 3D point when exploiting the prior. The complexity is comparable to that of SoftPOSIT [2].

A. Algorithm Overview

We extend the Blind PnP algorithm for line features. Given M 3D object lines, N 2D image line features (note that their corresponding relationship is not known), and a set of weak prior pose hypotheses with associated uncertainties, our aim is to find the correct camera pose (including three rotation angles and three translations) and the valid correspondences between the object lines and image lines. Our solution follows the framework of the Blind PnP algorithm. In more detail, for each initial pose guess with mean $\mathbf{p}_g$ and covariance Σ_g^p, we start a recursive loop to find the optimal pose and 3D-to-2D correspondences that minimize a defined matching objective error. In the recursive search, the current pose estimate and its uncertainty are used to restrict the number of plausible 3D-to-2D matches (Subsection 4.B). For every such match that is hypothesized, the pose with its uncertainty is updated in the way of a Kalman filter (Subsection 4.C) and is used to initialize the next depth of recursion. Once the pose and its covariance have been updated four times, i.e., four 3D-to-2D matches have been hypothesized, the uncertainty would become small enough that we can obtain the final pose estimate. We then evaluate the estimate's quality using the objective function (Subsection 4.D). After the evaluations are carried out over all plausible quadruples of correspondences, the best solution is retained. The method considers the possibility that the 2D feature lines of some 3D lines may not be detected

by postponing the making of hypotheses. When the probability of consecutively picking r nondetected 3D lines drops below a fixed threshold, the considering of 3D lines as nondetected is stopped. Our algorithm is summarized in Algorithm 1.

Algorithm 1: Algorithm of Simultaneous Pose and Correspondence from Lines

input: M 3D lines $\{L_i|i=1,2,\ldots,M\}$, N 2D image lines $\{l_j|j=1,2,\ldots,N\}$, G prior pose Gaussian components: $\{\mathbf{p}_g, \sum_g^p | g1,2,\ldots,G\}$.

output: Best pose estimate $\mathbf{p}_{\text{best}}$ and the valid 3D-to-2D line matches.

```
1 p_best ← p_1, minErrror ← ∞;
2 for g ← 1 to G do
3     p ← p_g, Σ^p ← Σ_g^p, nDepth ← 0, nHypotheses ← 0,
        nOutlier ← 0,
4     list ← {1,2,...,M}, p_est ← p_g, error → ∞;
5     RecursiveSearch (p, Σ^p, nDepth, nHypotheses, list,
        p_est, error, nOutlier);
6     if error < minError then
7         minError ← error;
8         p_best ← p_est;
9 find the valid matches for p_best;
// Define the function
RecursiveSearch.RecursiveSearch(p, Σ^p, nDepth,
nHypotheses, list, p_est, error, nOutlier){
10 if nDepth ≥ MaxDepth||nHypotheses ≥ 4 then
11     return
12 if nHypothesis = 4 then
13     tmpError ← ObjectiveFunction(p); // by Eq. (11)
14     if error > tmpError then
15         error ← tmpError; p_est ← p;
16     return;
17 if Consecutive_Outlier_Probability (nOutlier) <
       probability_threshold then
18     return;
19 for each i ∈ list do
20     find a set of 2D line match candidates S_i for L_i using
         Eq. (7);
21 find the set S_k from {S_i|i ∈ list} that has minimum number of
2D match candidates;
22 list removes k;
23 nDepth* ← nDepth + 1;
// case 1: this line L_k indeed has its 2D match.
24 for each j ∈ S_k do
25     p* ← KalmanUpdateMean(p, l_j); // by Eq. (9)
26     Σ^*p ← KalmanUpdateCov(Σ^p, l_j); // by Eq. (10)
27     nHypotheses* ← nHypotheses + 1;
         nOutlier* ← nOutlier;
28     RecursiveSearch(p*, Σ^*p, nDepth*, nHypotheses*,
         list, p_est, error, nOutlier*);
// case 2: this line L_k has its 2D match not detected.
29 nHypotheses* ← nHypothese; nOutlier* ← nOutlier + 1;
30 RecursiveSearch(p, Σ^p, nDepth*, nHypotheses*, list, p_est,
   error, nOutlier*);}
```

B. Line Projection Uncertainty for Finding of Potential Matches

For each prior pose with mean $\mathbf{p}_g$ and covariance Σ_g^p, we verify it by generating a small subset of 3D-to-2D correspondences using the uncertainty information. We estimate the uncertainty associated with the projection of a 3D object line. The estimated uncertainties are then used to predict the bounds of the projections and restrict the possibility of matches between the 3D and 2D lines.

A 2D straight line in the image can be represented by a vector $\mathbf{l}_i$ of two independent variables, for example, in Hough space. Without loss of generality, here we define $\mathbf{l}_i = (k_i, b_i)^T$ and use the form $k_i u + v + b_i = 0$ (k_i and b_i cannot both be 0) to represent a 2D line in pixel coordinates. Let $\mathbf{l}_i$ denote the 2D projection of a 3D line L_i. We can compute the projection $\mathbf{l}_i$ and its uncertainty through the projections of the two endpoints of the 3D line. For two 3D endpoints $\mathbf{P}_i^1$ and $\mathbf{P}_i^2$ of L_i, their corresponding 2D points in pixel coordinates $\mathbf{u}_i^1 = (u_1, v_1)^T$ and $\mathbf{u}_i^2 = (u_2, v_2)^T$ are obtained from the perspective projection: $\mathbf{u}_i^j = f_{\text{proj}}(\mathbf{p}_g, \mathbf{P}_i^j), j = 1, 2$. $\mathbf{l}_i$ is then computed by

$$\mathbf{l}_i = f_l(\mathbf{u}_i^1, \mathbf{u}_i^2) = \left(\frac{v_2 - v_1}{u_1 - u_2}, \frac{v_1 u_2 - v_2 u_1}{u_1 - u_2}\right)^T, \quad (4)$$

and the projection uncertainty of L_i, by

$$\Sigma_i = \boldsymbol{J}(\mathbf{u}_i^1, \mathbf{u}_i^2)\Sigma_i^*(\boldsymbol{J}(\mathbf{u}_i^1, \mathbf{u}_i^2))^T, \quad (5)$$

where $\boldsymbol{J}(\mathbf{u}_i^1, \mathbf{u}_i^2)$ is the Jacobian of f_l and Σ_i^* is a diagonal covariance matrix $\Sigma_i^* = \text{diag}(\Sigma_i^1, \Sigma_i^2)$, with Σ_i^j as the covariance of 3D points $\mathbf{P}_i^j$:

$$\Sigma_i^j = J(\mathbf{P}_i^j)\Sigma_g^p(J(\mathbf{P}_i^j))^T, \quad (6)$$

where $J(\mathbf{P}_i^j)$ is the Jacobian of the perspective projection function f_{proj}.

The uncertainty Σ_i predicts the bounds of the line projection and only those image lines $\mathbf{l}_j$, such that

$$(\mathbf{l}_i - \mathbf{l}_j)^T(\Sigma_i)^{-1}(\mathbf{l}_i - \mathbf{l}_j) \le \mathbb{M}^2 \quad (7)$$

are considered as the potential matches for the 3D line L_i. $\mathbb{M}$ set as 2 or 3 yields 86% or 99% of confidence, respectively. For every 3D line L_i ($i = 1, 2, \ldots M$), we now can find a set of 2D line match candidates $\mathbb{S}_i$ from Eq. (7).

C. Pose and Uncertainty Update

The sets $\mathbb{S}_i$ ($i \in \{1, 2, \ldots, M\}$) cover the most likely 2D match hypotheses for every 3D line L_i given the initial pose $\mathbf{p}_g$ and its uncertainty Σ_g^p. Among these sets, we find the one $\mathbb{S}_k$ that has the minimum number of match candidates and ignore all others. The later search or update of the pose and correspondences will be based only on this set. The set $\mathbb{S}_k$ corresponds to $|\mathbb{S}_k|$ ($\|$ denotes the size of a set) pairs of match hypotheses: $\{(L_k, l_j)|j = 1, 2, \ldots, |\mathbb{S}_k|\}$. Since we do not know which pair is better than the others, we have to consider each one in turn. The establishment of one pair match (L_k, l_j) between the 3D line L_k and image line l_j would result in the reduction of uncer-

tainty in the pose and its covariance. Hence, with such a match hypothesis we can update the pose and its uncertainty, and we utilize Kalman filter for this task.

Given the variable $\mathbf{l}_j = (k_j, b_j)^T$ of line l_j, we define $\mathbf{m}_j = (k_j, 1, b_j)^T$ and A as the matrix of camera intrinsic parameters. We obtain a constraint equation of $\mathbf{m}_j$ according to Eqs. (1), (2), and (3):

$$f(\mathbf{l}_j, \mathbf{p}_g) = \begin{bmatrix} \mathbf{m}_j A R d_i \\ \mathbf{m}_j A (R p_i + \mathbf{t}) \end{bmatrix}. \tag{8}$$

If l_j is the correct match of line L_i, $f(\mathbf{l}_j, \mathbf{p}_g) = 0$ holds. We now can update the pose and its covariance according to the following Kalman filter equations:

$$\mathbf{p}_g^* = \mathbf{p}_g - Qf(\mathbf{l}_j, \mathbf{p}_g), \tag{9}$$

$$\Sigma_g^{*p} = (I - Q\mathbf{M})\Sigma_g^p, \tag{10}$$

where

$$Q = \Sigma_g^p \mathbf{M}^T (\mathbf{U} + \mathbf{M}\Sigma_g^p \mathbf{M}^T)^{-1}, \mathbf{M} = \frac{\partial f}{\partial \mathbf{p}_g},$$
$$\mathbf{U} = \frac{\partial f}{\partial \mathbf{l}_j} \mathbf{U}^* \left(\frac{\partial f}{\partial \mathbf{l}_j}\right)^T,$$

and $\mathbf{U}^*$ is the covariance of noise generated mainly from image sensor and feature detection.

After the update of the pose and its covariance, the new estimation is used to find the second match hypothesis from the remaining 3D lines. This step is recursively repeated until at least three hypotheses are found (in our experiment we set the number to be four). After four updates, the pose covariance would become very small. The quality of the estimation is then evaluated by a matching error metric.

The above assumes that the line L_k has the correct 2D match. However, it is possible that it has not detected its 2D match. We handle this case by skipping this line and continue on to consider the next best 3D line for the next match hypothesis. We do not need to consider this case for every line, since the possibility of consecutively picking r nondetected lines decreases quickly when r increases.

The search process is repeated for each Gaussian component, and the best one is retained. In the next section, we define the matching evaluation function and describe the method of finding the optimal correspondences for an estimated pose.

D. Matching Objective Error and the Finding of Optimal Correspondences

For a pose $\mathbf{p}$ that has been attained, we evaluate its quality by using an objective function. Assuming that a one-to-one match can be established for K lines among the total M 3D object lines with no match for the remaining $M - K$ lines, we define an objective function as follows:

$$E(\mathbf{p}) = \sum_{i=1}^{K} e_{ij} + (M - K)\mathbb{T}, \tag{11}$$

where

$$e_{ij} = w_1 \|\mathbf{n}_j^T R \mathbf{d}_i\|^2 + w_2 \|\mathbf{n}_j^T (R\mathbf{P}_i + \mathbf{t})\|^2. \tag{12}$$

The function e_{ij} evaluates the quality of the match between 3D line L_i and image line l_j. The constant weight factors w_1 and w_2 balance the error influence between the two constraints (Section 3). $\mathbb{T}$ is a penalty value that penalizes unmatched lines.

Finding the optimal one-to-one correspondences for the remaining 3D lines is performed by two steps. For every remaining 3D line L_i, we first find the set of potential 2D match candidates by the method described in Subsection 4.B. Then we exhaustively find the one l_k that has the minimum error of e_{ij}, i.e.,

$$k = \arg_j \min e_{ij}. \tag{13}$$

If $e_{ik} < \mathbb{T}$, we then assign l_k as the match of L_i; otherwise, it is considered as not detected. $\mathbb{T}$ is set to $9\sigma^2$ (σ^2 is the noise variance) to ensure with 99% probability that a 3D line and a 2D line actually match.

5. Pose from Known Line Correspondences

After the simultaneous estimation of both pose and correspondences, a sophisticated pose estimation algorithm may be needed to refine the attained pose. This is because the hypothesize-and-test strategy estimates the pose using a subset of match hypotheses, leading to a final pose that may not be sufficiently accurate. Given the established correspondences, a pose estimation algorithm that computes the pose from the whole set of correspondences can provide a global optimal estimate. On the other hand, most existing iterative line based pose estimation algorithms may not fully exploit the specific structure of the problem of pose estimation from line features and are usually sensitive to the initial noise or outliers. Hence we are motivated to develop a new algorithm of pose estimation from line correspondences.

A. Object-Space Objective Function

The fundamental equations indicate that each 3D-to-2D line correspondence provides two constraints. If N line correspondences are available, the pose problem becomes the problem of minimizing the following objective function [25–27]:

$$E(R, \mathbf{t}) = \sum_{i=1}^{N} (\mathbf{n}_i^T R \mathbf{d}_i)^2 + \sum_{i=1}^{N} (\mathbf{n}_i^T (R\mathbf{P}_i + \mathbf{t}))^2. \tag{14}$$

Instead of using the objective function of Eq. (14), we present and use an objective metric based on the coplanarity error in the object space. In fact, similar to thinking of collinearity of point correspondence as orthogonal projection [24], the coplanarity of a

3D-to-2D line correspondence can be understood as the projection of the 3D line in the interpretation plane coinciding with itself. In the camera coordinate system, the line direction vector is $R\mathbf{d}_i$, and its projection in the interpretation plane can be obtained by $(I-\mathbf{n}_i\mathbf{n}_i^T)R\mathbf{d}_i$, where I is a 3×3 identity matrix. Letting $K_i = I-\mathbf{n}_i\mathbf{n}_i^T$, we have

$$R\mathbf{d}_i = K_i R\mathbf{d}_i. \tag{15}$$

Similarly, for the point position vector $R\mathbf{P}_i+\mathbf{t}$, we have

$$R\mathbf{P}_i+\mathbf{t} = K_i(R\mathbf{P}_i+\mathbf{t}). \tag{16}$$

From Eqs. (15) and (16), we have the corresponding vector differences as

$$\mathbf{e}_i^d = (I-K_i)R\mathbf{d}_i, \tag{17}$$

$$\mathbf{e}_i^P = (I-K_i)(R\mathbf{P}_i+\mathbf{t}). \tag{18}$$

We refer to $\mathbf{e}_i^d$ and $\mathbf{e}_i^P$ as coplanarity errors. We then achieve the following objective functions:

$$E_1(R) = \sum_{i=1}^{N}\|\mathbf{e}_i^d\|^2 = \sum_{i=1}^{N}\|(I-K_i)R\mathbf{d}_i\|^2, \tag{19}$$

$$E_2(R,\mathbf{t}) = \sum_{i=1}^{N}\|\mathbf{e}_i^P\|^2 = \sum_{i=1}^{N}\|(I-K_i)(R\mathbf{P}_i+\mathbf{t})\|^2. \tag{20}$$

Compared with Eq. (14), the two objective functions [Eqs. (19) and (20)] have clear geometry, meaning that the optimal solution of the pose should achieve a minimum value of the sum of the squares of the coplanarity errors (see Fig. 2).

Note that like the line-of-sight projection matrix V_i defined in [24], K_i is a interpretation plane projection matrix that, when applied to an object vector, projects the vector orthogonally to the interpretation plane. It satisfies the following properties:

$$\|K_i\| \geq \|K_i\mathbf{x}\|, \qquad \mathbf{x}\in\mathbb{R}^3, \tag{21}$$

$$K_i^T = K_i, \tag{22}$$

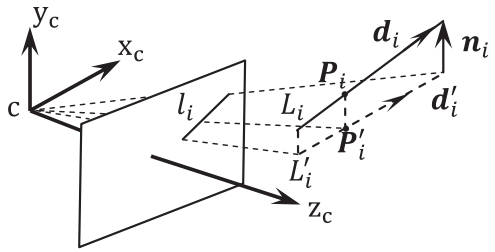


Fig. 2. Object-space error of line correspondence.

$$K_i^2 = K_iK_i^T = K_i. \tag{23}$$

B. New Algorithm

We propose a pose estimation algorithm that alternatively optimizes the rotation matrix and translation vector. Before describing the details of our new pose estimation method, we show a lemma given by [29]:

Lemma 1: Let A and B be $m\times n$ matrices, R an $m\times m$ rotation matrix, and UDV^T a singular value decomposition (SVD) of AB^T ($UU^T=VV^T=I$, $D=\mathrm{diag}(\lambda_i)$, $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_m\geq 0$). To minimize the objective function

$$f(R) = \|A-RB\|^2, \tag{24}$$

the optimal rotation matrix R must be such that

$$R = USV^T. \tag{25}$$

When $\mathrm{rank}(AB^T) > m-1$, S must be chosen as

$$S = \begin{cases} I & \text{if} \quad \det(AB^T)\geq 0, \\ \mathrm{diag}(1,1,\ldots,1,-1) & \text{else.} \end{cases} \tag{26}$$

When $\det(AB^T)=0$ ($\mathrm{rank}(AB^T)=m-1$), S must be chosen as

$$S = \begin{cases} I & \text{if} \quad \det(U)\det(V)=1, \\ \mathrm{diag}(1,1,\cdots,1,-1) & \text{if} \quad \det(U)\det(V)=-1. \end{cases} \tag{27}$$

Lemma 1 gives us a solution to the optimal estimate of rotation matrix R. Let us assume we are given N 3D-to-2D line correspondences and we have obtained the projection operator K_i for each correspondence. We define the following:

$$A = (K_1R\mathbf{d}_1, K_2R\mathbf{d}_2, \ldots, K_NR\mathbf{d}_N),$$
$$B = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N).$$

$E_1(R)$ then becomes

$$E_1(R) = \|A-RB\|^2. \tag{28}$$

It is seen that Eq. (28) bears a close resemblance to Eq. (24). We can now obtain an estimate of the rotation according to Eq. (25).

After attaining an estimate rotation R, the optimal estimate of translation $\mathbf{t}$ can be computed easily by minimizing $E_2(R,\mathbf{t})$ of Eq. (20):

$$\mathbf{t} = t(R) = \left(\sum_{i=1}^{N}(I - K_i)\right)^{-1} \sum_{i=1}^{N}(K_i - I)R\mathbf{P}_i. \quad (29)$$

Clearly the matrix $(\sum_{i=1}^{N}(I - K_i))$ must be invertible for Eq. (29) to hold. Since $I - K_i = \mathbf{n}_i\mathbf{n}_i^T$, we have $(\sum_{i=1}^{N}(I - K_i)) = CC^T$, where

$$C = \begin{bmatrix} a_1 & a_2 & \cdots & a_N \\ b_1 & b_2 & \cdots & b_N \\ c_1 & c_2 & \cdots & c_N \end{bmatrix}^T.$$

Hence, if $\text{rank}(CC^T) \equiv 3$, Eq. (29) can be well defined. In fact, $\text{rank}(CC^T) = \text{rank}(C) = 3$ is always true if $N \geq 3$ and the N interpretation planes do not all intersect in one line. In other words, if there are at least three of the 3D lines that do not intersect in one point, the value of translation $\mathbf{t}$ can always be computed by Eq. (29).

With the above preparations, we can derive our algorithm now. Assuming we have obtained the kth estimation of R_k, we first estimate R'_{k+1} from Eq. (25) and obtain a new estimate of translation $\mathbf{t}'_{k+1}$ via $t(R'_{k+1})$ from Eq. (29). Then with $(R'_{k+1}, \mathbf{t}'_{k+1})$, we use the method of [9] to obtain the final $(k+1)$th estimation by minimizing the objective function $E_2(R, \mathbf{t})$. The last step is described as follows.

In the algorithm of [9], R and $\mathbf{t}$ are iteratively optimized by minimizing an object-space objective function defined for the point correspondences:

$$E(R, \mathbf{t}) = \sum_{i}^{N} \|(I - V_i)(R\mathbf{P}_i + \mathbf{t})\|^2, \quad (30)$$

where V_i is a projection matrix that, when applied to a point, projects the point orthogonally to the line of sight defined by the image point. When R^k and $\mathbf{t}^k$ are obtained, the next estimate R^{k+1} is determined by solving the following absolute orientation problem:

$$R^{k+1} = \underset{R}{\text{argmin}} \sum_{i=1}^{N} \|R\mathbf{P}_i + \mathbf{t} - V_i\mathbf{q}_i^k\|^2, \quad (31)$$

where $\mathbf{q}_i^k = R^k\mathbf{P}_i + \mathbf{t}^k$. This absolute orientation problem is then solved by the SVD method [30].

It is seen that the only difference in Eq. (20) compared with Eq. (30) is the use of projection matrix K_i instead of V_i. Both projection vectors K_i and V_i bear the same properties (Subsection 5.A). Hence, after we have obtained the estimates $(R'_{k+1}, \mathbf{t}'_{k+1})$, an estimate of rotation, R_{k+1}, can be computed by directly using the algorithm of [9] to minimize the objective function (20). The new algorithm is summarized in Algorithm 2. We name the new algorithm the line based orthogonal iteration (LOI) algorithm.

Algorithm 2: Line Based Orthogonal Iteration (LOI) Algorithm

input: $N (N \geq 3)$ 3D-to-2D line correspondences: $\{(L_i, l_i) | i = 1, 2, \ldots, N\}$, with 3D line L_i represented by $\lambda\mathbf{d}_i + \mathbf{P}_i$ and 2D detection l_i by $a_i x + b_i y + c_i = 0$, where $\mathbf{n}_i = (a_i, b_i, c_i)^T$; initial pose rotation R_0 and translation $\mathbf{t}_0$.
output: The optimal rotation and translation.
1 Define $K_i = I - \mathbf{n}_i\mathbf{n}_i^T$, $B = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N)$, $\mathbf{P}'_i = \mathbf{P}_i - \bar{\mathbf{P}}_i$ where $\bar{\mathbf{P}}_i = \frac{1}{N}\sum_{i=1}^{N}\mathbf{P}_i$; Set $k = 0$;
2 **while** *not converge* **do**
3 $A \leftarrow (K_1R_1\mathbf{d}_1, \ldots, K_NR_k\mathbf{d}_N)$;
4 $M \leftarrow AB^T$ and perform SVD: $UDV^T \leftarrow M$;
5 $R'_{k+1} \leftarrow USV^T$, where S is set according to Eqs. (26) and (27);
6 $\mathbf{t}'_{k+1} \leftarrow t(R'_{k+1})$;
7 // Given R'_{k+1} and $\mathbf{t}'_{k+1}$, compute R_{k+1} as follows:
8 (1) **for** each i, define $\mathbf{q}_{i,k+1} = K_i(R'_{k+1}\mathbf{P}_i + \mathbf{t}'_{k+1})$,
9 $\bar{\mathbf{q}}_{i,k+1} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{q}_{i,k+1}$, $\mathbf{q}'_{i,k+1} = \mathbf{q}_{i,k+1} - \bar{\mathbf{q}}_{i,k+1}$;
10 (2) $\mathbf{M}'(R) \leftarrow \sum_{i=1}^{N}\mathbf{q}'_{i,k+1}\mathbf{P}'^T_i$;
11 (3) perform SVD; $U'D'V'^T \leftarrow M'(R)$;
12 (4) $R_{k+1} \leftarrow V'U'^T$;
13 (5) $\mathbf{t}_{k+1} \leftarrow t(R_{k+1})$;
14 $k = k + 1$;

6. Experiments

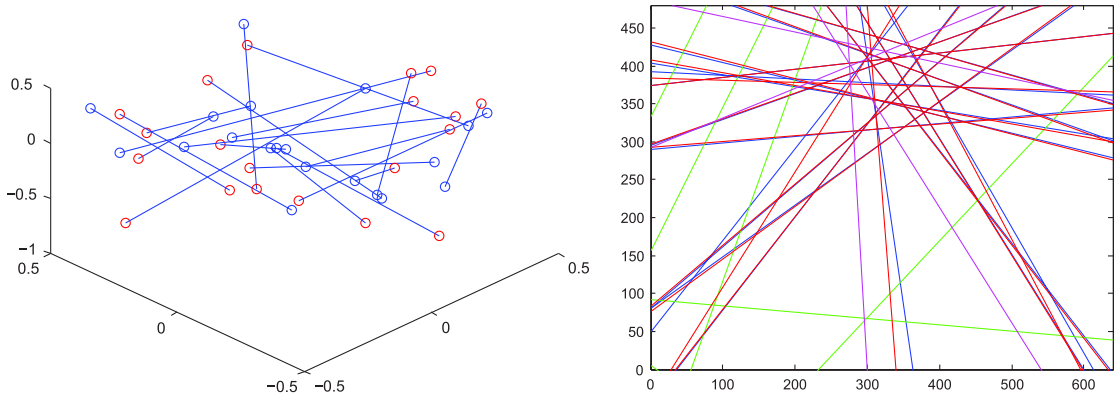
To evaluate our methods, we carried out experiments on both synthetic and real data.

A. Synthetic Data

1. *Simultaneous Pose and Correspondence*

We first evaluate our approach to the simultaneous pose and correspondence problem through extensive Monte Carlo simulations. In each simulation, the effects of clutter, occlusion, and noise are considered. Pose priors are modeled by a 20-component GMM, which can be learned from thousands of uniformly generated pose training data [4]. To generate the input data for each trial, we first generate a set of points randomly located in a sphere centered at the object's origin, then connect each pair of two random points that are not too close as a 3D line. Let M denote the total number of 3D lines. To generate the 2D image lines, we first sample a number of points, e.g., $N_s = 100$, from each 3D line. By taking a randomly selected pose from the training pose data as the true pose, we project the sampling 3D points to a 640×480 image with a defined virtual camera, of which the focal length $f = 800$, with the principal point at the center of the image. Gaussian noise with mean zero and standard deviation σ is then added to the projection points. Then M 2D lines are generated by performing linear fitting on the corresponding projection points. We delete randomly selected image lines to simulate the occlusion and add a number of image lines to simulate the clutter. Let p_o and p_c denote the ratio of occlusion and clutter, respectively. Then Mp_o 2D occlusion lines must be deleted and $M(1 - p_o)p_c/(1 - p_c)$ clutter lines must be added. Figure 3 shows one typical example of the input data simulated in our experiments.

Figure 4 shows a typical simulation of running our algorithm to find the correct pose and correspondences. In such a trial, we record the updates of the best estimate attained by the search using each

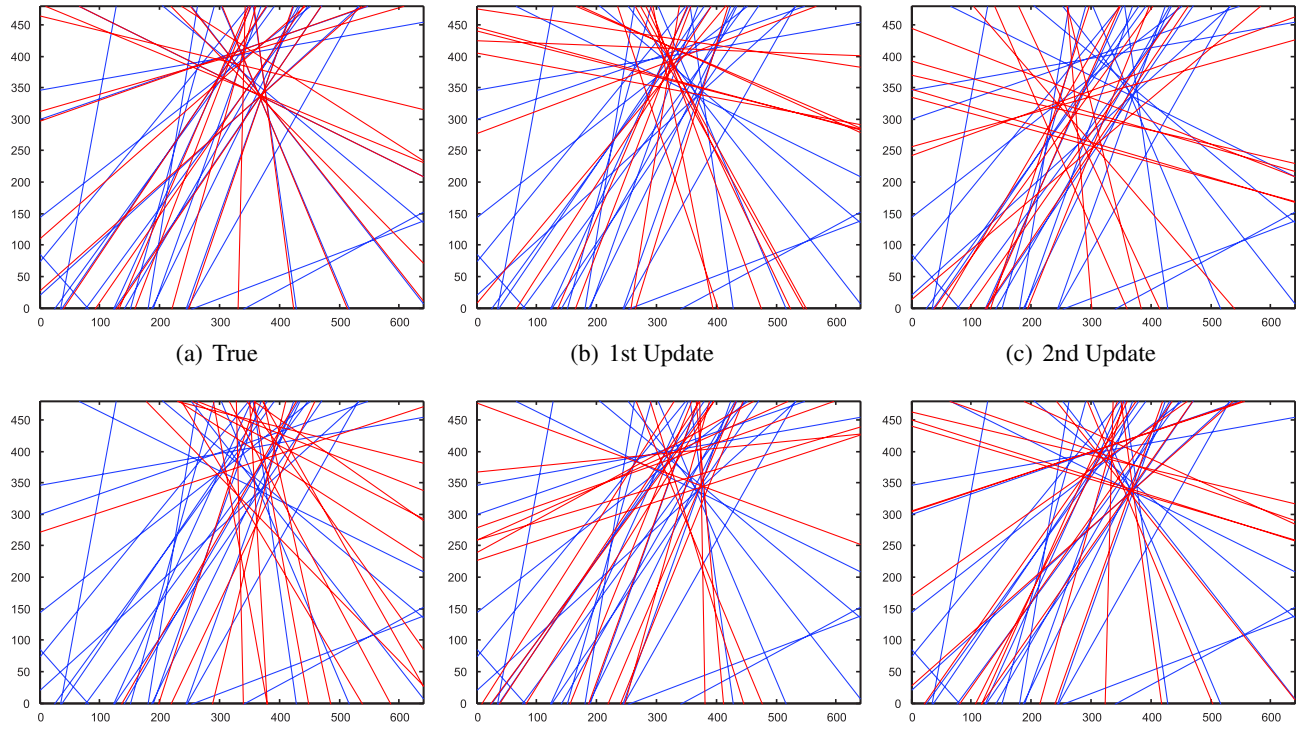


(a) (b)

Fig. 3. (Color online) (a) Example of the simulation of the 3D object lines and (b) the detected infinite 2D image lines. The 3D lines are generated in 3D space by connecting two random endpoints that are not too close. Their projections are drawn in either red or pinkish red. The red ones are those that have true detected 2D lines, while the pinkish red ones are those have no true detections that may caused by occlusions. The detected 2D image lines are either in blue or green. Blue denotes nonclutter, while green means clutter detections. Gaussian noise is added to perturb the locations of the endpoints of each 2D line. Here the noise deviation $\sigma = 2$ pixels.

prior pose component of the GMM. We take a trial as success or convergence if the relative rotation error e_r and translation error t_r are both less than 0.15, where $e_r = \frac{\|R_{est}\|}{\|R_{true} - R_{est}\|}$ and $\mathbf{t}_r = \frac{\|\mathbf{t}_{est}\|}{\|\mathbf{t}_{true} - \mathbf{t}_{est}\|}$. Figure 5 shows the success rates as a function of the number of 3D lines and a set of combinations of p_o and p_c. One hundred trials were run for each experimental case.

We empirically analyzed the computational complexity of our algorithm by comparing it with the Blind PnP method [4]. Figure 6 shows the computational time as a function of the number of object lines.



(a) True (b) 1st Update (c) 2nd Update

(d) 3rd Update (e) 4th Update (f) 5th Update

Fig. 4. (Color online) Example of using our simultaneous pose and correspondence method on a synthetic cluttered image. In all simulated images (image size is 640 × 480), blue lines are the 2D detected lines, among which there are some clutter lines; red lines are the projections of all the 3D lines at the current estimated pose. (a) True pose, (b)–(f) successive pose update steps. The correct pose is found by five update steps. The experimental parameters: $N = 20$, $p_o = 0.2$, $p_c = 0.4$, and $\sigma = 2$.

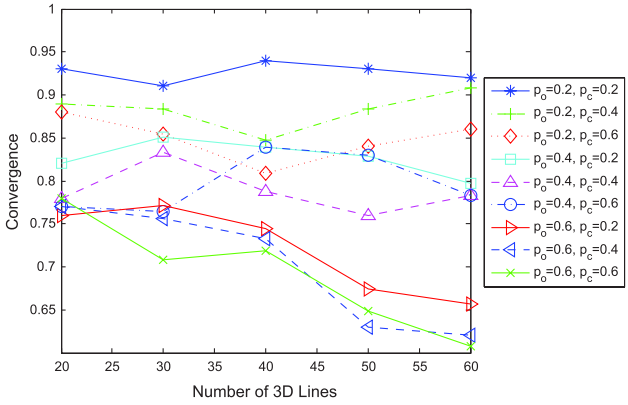


Fig. 5. (Color online) Convergence percentage for different combinations of settings of occlusion p_o and clutter p_c. The number of object lines varies from 20 to 60.

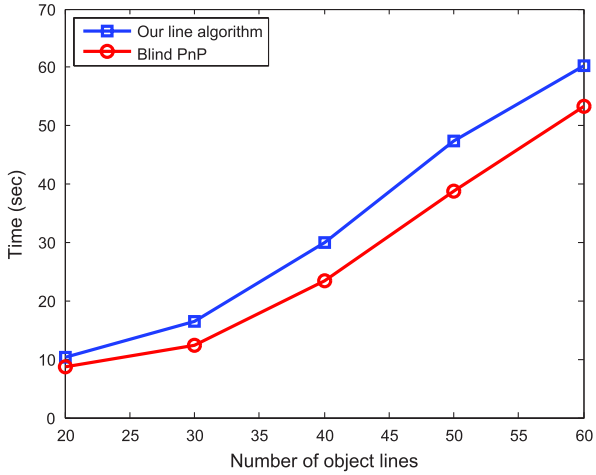


Fig. 6. (Color online) Computational time as a function of the number of 3D object lines. The occlusion percentage $p_o = 0.4$, and the clutter percentage $p_c = 0.6$.

For each trial, M 3D lines are used for our line algorithm and the randomly selected M endpoints are used for the BlindPnP method. It is seen that the computation times of our line algorithm are comparable to those of the Blind PnP method, of which the complexity is reported to be comparable to that of SoftPOSIT [2]. Since the extension of use of SoftPOSIT to line features [3] as shown has the same runtime complexity as the SoftPOSIT algorithm for point features, we may say that the computational complexity of our line method can be comparable to SoftPOSIT for line features, although we did not quantitatively compare the two algorithms.

2. *Pose Estimation*

In the simulations carried out to evaluate our pose estimation algorithms for line features, the input data is generated similarly to Subsection 6.A.*1*. A set of 3D lines are generated uniformly within a cube defined by $[-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$ (Fig. 3) in the object space. The corresponding 2D lines are then created by linear fitting of the projections of a set of sampling points in the 3D lines. We add Gaussian noise to the projections of points and also consider a percentage p_{out} of outliers, for which a set of 3D lines are randomly selected and replaced by another line generated within the cube $[-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$. For each setting of the control parameters in every plot, the result is obtained by running 1000 trials and the mean value is recorded.

The proposed LOI algorithm uses an alternative optimization framework. First an estimate is obtained by minimizing the constraint of line direction [e.g., the objective function of Eq. (19)], then the result is used to initialize a further optimization on the line position constraint [e.g., the objective function of Eq. (20)]. This two-stage optimization strategy fully explores the line constraint information. To study the performance of our alternative optimization based LOI algorithm, we compare it with two other strategies that separately optimize the two objective functions. The first one iteratively minimizes the objective function of Eq. (19) to obtain the final rotation estimation, i.e., iteratively optimize rotation via Eq. (25), and then computes the translation via $t(R)$. The second strategy is to carry the optimization only on objective function Eq. (20). Given an initial estimate for both R and $\mathbf{t}$, this method purely minimizes $E_2(R, t)$ by using the method of [9]. For convenience, we name the two methods direction optimization

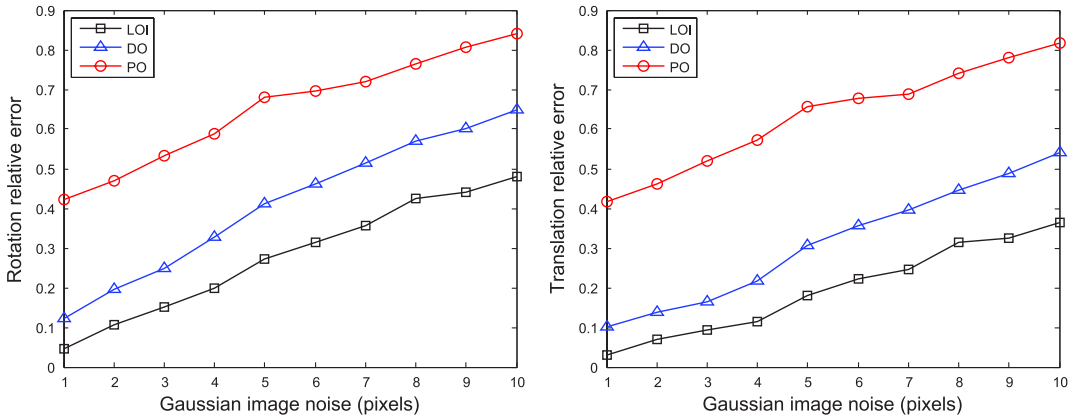


Fig. 7. (Color online) Relative rotation and translation error as a function of image noise when the number of lines is fixed to be 8.

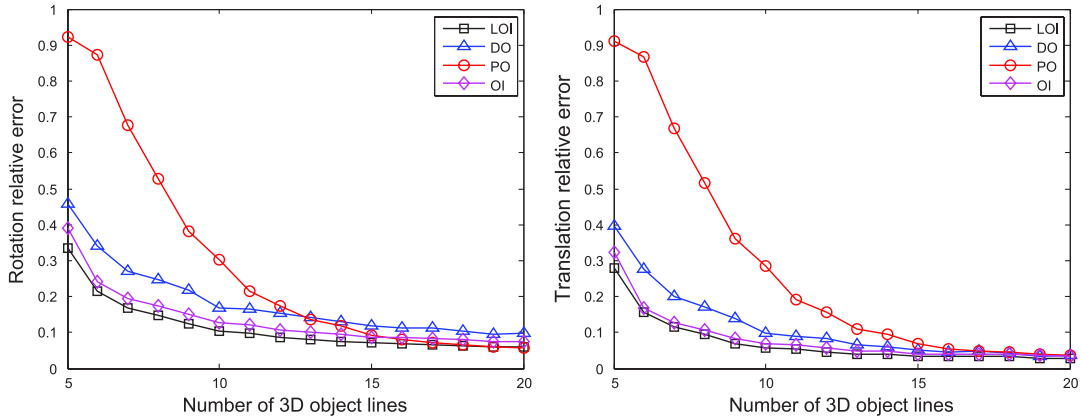


Fig. 8. (Color online) Relative rotation and translation error as a function of the number of object lines when the standard deviation of image noise is fixed to be 3 pixels.

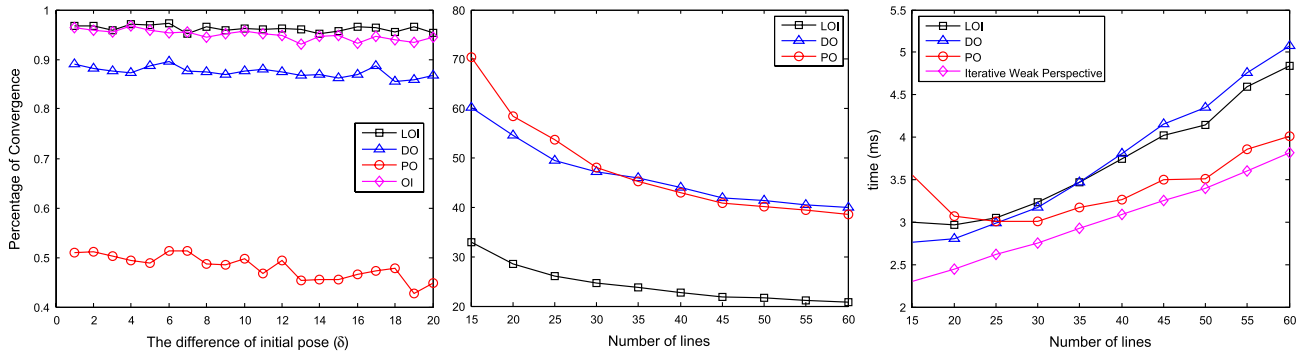


Fig. 9. (Color online) (a) Percentage of convergence when initial poses are generated from a multinormal distribution, (b) the number of iterations, and (c) the running time as a function of the number of lines.

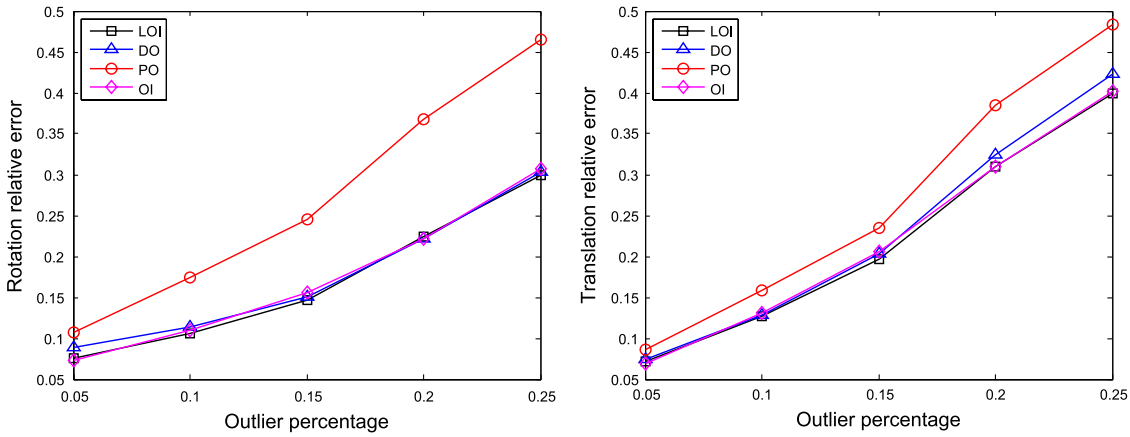
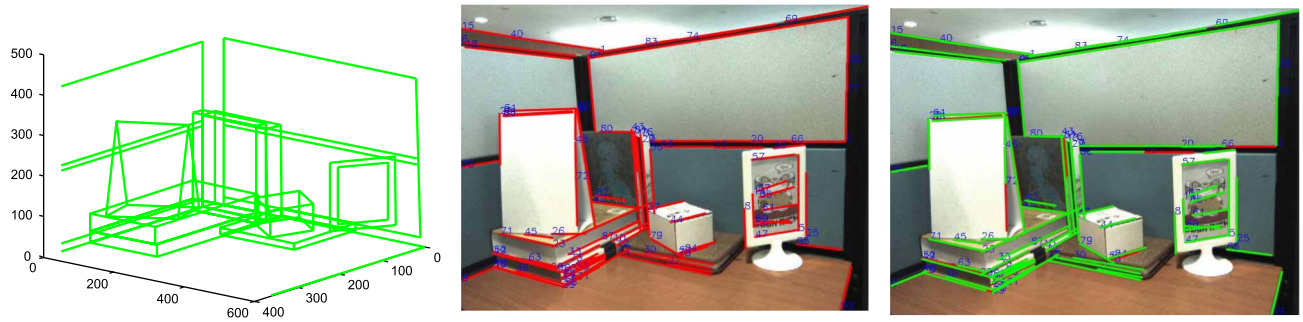


Fig. 10. (Color online) Relative rotation and translation error as a function of the percentage of outliers. For all trials, the number of object lines is 25 and the standard deviation of image noise is 3.

(DO) and position optimization (PO), in contrast to our alternative optimization of LOI algorithm.

In Fig. (7), we plot the rotation and translation relative errors produced by the three algorithms as a function of Gaussian noise with its standard deviation varying from 1 to 10 pixels. The number of sampling points that are used for creating the 2D lines is set as 100. The line number is fixed to be 8, and the percentage of outliers $p_{out} = 0$. The plots show that LOI is consistently more accurate than the other two algorithms.

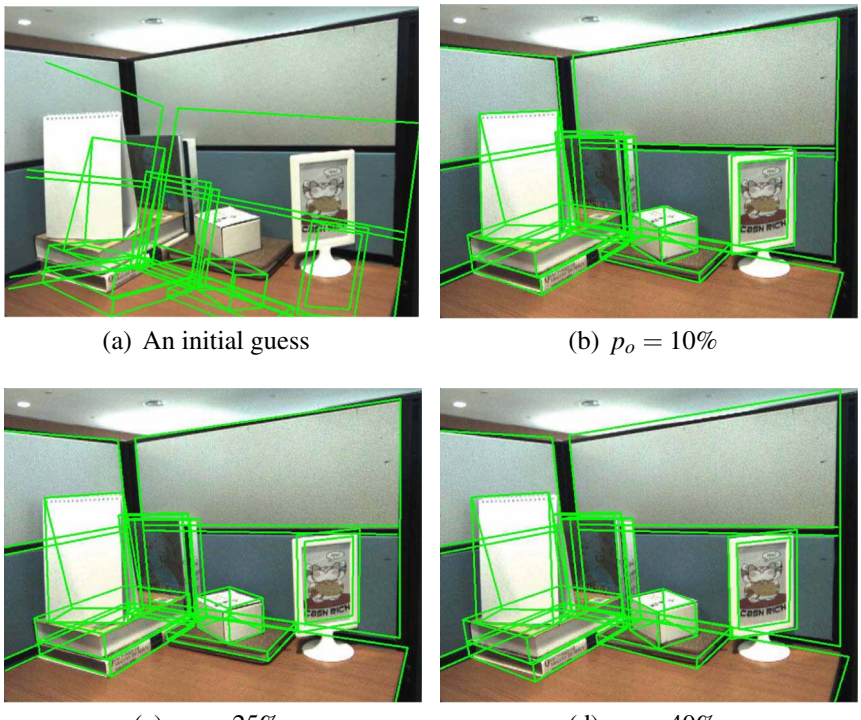


(a) (b) (c)

Fig. 11. (Color online) (a) 3D object model; (b) 2D lines are extracted by Hough transform; (c) a number of clutter lines or repeat detections are removed to limit the number of valid lines to between 50 and 80 (green lines are the final detections). Note that although the lines are detected as segments, they are used as infinity lines in all algorithms.

Figure 8 plots the error as a function of the number of 3D object lines when the image noise is fixed ($\sigma = 3$ pixels). We compare the performance of our LOI algorithm with the OI method [9]. For a relatively fair comparison, if m 3D lines are used for LOI, $2m$ corresponding endpoints of the 3D lines are used for OI algorithm. Since two endpoints determine one line in the space, the input for the point OI algorithm and other line algorithms is similar. It can be seen that all these algorithms can achieve higher accuracy when the number of feature correspondence increases. LOI and OI algorithms show more accurate and stable performance. Two reasons may cause the comparable performance between OI and LOI algorithm. One is the similar input setting for both algorithms, and another is that both algorithms perform optimization by fully exploiting the orthogonal projection constraints.

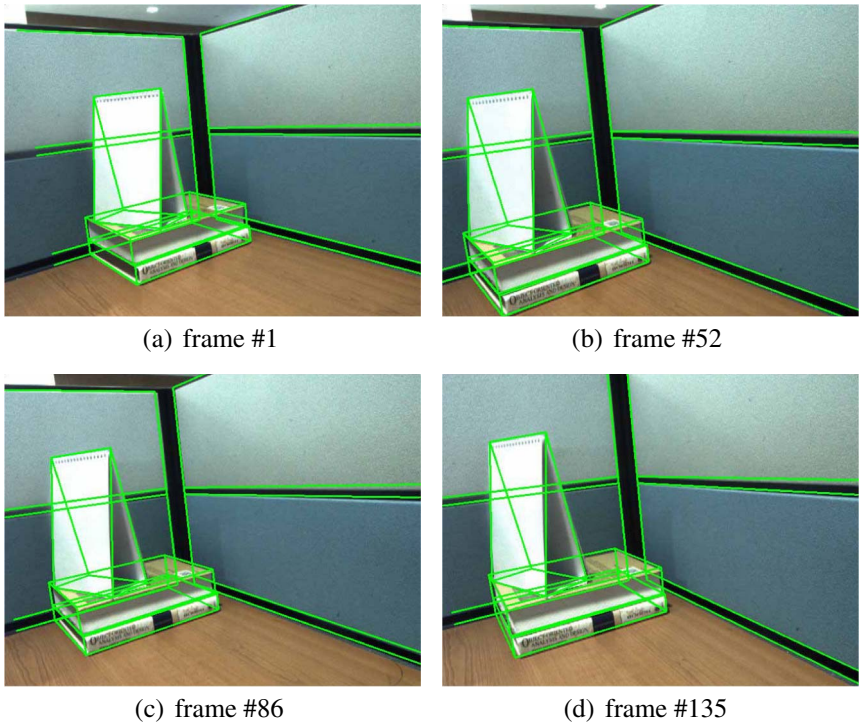
In Fig. 9(a), we give the percentage of convergence when the initial poses are generated from a multinormal distribution with the mean as the true pose and the diagonal covariance $\delta\Sigma_0$, where the standard deviation element of Σ_0 is about 1.5 deg for the rotation angles, 0.2 for the x and y components of the translation t, and 0.5 for the z component. δ varies from 1 to 20. The plot indicates that the LOI algorithm shows very robust performance and slightly outperforms the OI algorithm, which is proved to be globally convergent. Figure 9(c) gives the computation times. We compare the LOI algorithm with the iterative weak



(a) An initial guess (b) $p_o = 10\%$

(c) $p_o = 25\%$ (d) $p_o = 40\%$

Fig. 12. (Color online) (a) Example of the mean pose of one prior Gaussian component, (b)–(d) example results for different occlusion cases.



(a) frame #1 (b) frame #52

(c) frame #86 (d) frame #135

Fig. 13. (Color online) Pose estimation for 3D object tracking on a sequence recorded with a moving camera.

perspective method [8], which efficiently estimates a pose with a weak perspective camera model and improves the estimation iteratively by solving an approximate system of linear equations.

Figure 10 plots the relative rotation and translation error as a function of the number of outliers. For all trials, the number of object lines is 25 and the standard deviation of the image noise is 3 pixels. For an outlier ratio p_{out}, Mp_{out} 3D lines are randomly replaced by other lines. This leads to the same outlier ratio for the point OI method, since the $2M$ endpoints are considered for OI. It is seen that LOI and OI algorithms show more accurate and stable performance.

B. Real Data

In this section, we validate our approaches by using real images of the scene shown in Fig. 11. This scene model consists of about 100 lines, of which a number of lines are occluded. We create its 3D model as shown in Fig. 11(a).

We first test our simultaneous pose and correspondence algorithm. Given an image of the scene, lines are first detected by using a Hough transform on the binary edge image obtained by Canny (Fig. 11). The number of detected lines used for the algorithms is limited between 50 and 80 by removing a number of lines that are found as clutter or repeated detections. In such a step, if one line is found parallel and too close to a line segment that has already been labeled, the line is removed. Although some clutter can be removed in this step, most clutter still exists. For example, in the case shown in Fig. 11(c), the percentage of clutter is about 35%. The pose prior is modeled as a Gaussian mixture with 10 components. Figure 12(a) shows an example of the mean pose of one prior Gaussian component. At any point of view of the camera, for example, the image shown in Fig. 12, there are always some lines that are occluded. We consider the occlusion by randomly selecting a fraction (p_o) of object lines that are known to be occluded and use them for running the trials. For each setting of the control parameters, we ran many trials to retain the best results when the total number of object lines is fixed to be 40. The results are displayed by projecting the CAD model to the camera image using the recovered pose, as shown in Figs. 12(b)–12(d).

We also validated our pose estimation approach for line correspondence, by using the algorithm for 3D line object tracking. The tracking is performed on an image sequence recorded from a moving calibrated camera pointing toward the scene, as shown in Fig. 13. We implement a simple line tracker in a typical framework where a local search along the normal direction of a model edge line is performed for the gradient maxima for a set of sampled points in the line [31]. The strong maxima are taken as the 2D feature points that correspond to the 3D sampled points in the object line. At run time, the tracker generates a set of 3D-to-2D line correspondences among which outliers or erroneous ones exist. A robust pose estimation method that resists outliers well is invaluable for robust tracking. We use the LOI algorithm for the tracking. Figure 13 shows four frames of the tracked sequence. Our method consistently tracks the whole sequence.

7. Conclusion

We have proposed two algorithms for the model-to-image registration problem of line features. Without knowing the 3D-to-2D line correspondences, we exploit weak pose priors to simultaneously estimate the camera pose and 3D-to-2D line correspondences. To verify the pose guesses, the method recursively finds a subset of 3D-to-2D line match hypotheses and progressively updates the pose using a Kalman filter. The use of pose priors and the exploiting of line features makes the algorithm robust to clutter and occlusion. We also present a new line based pose estimation algorithm (LOI) for the situation where the correspondences are given or have been established by the simultaneous pose and correspondence methods. Our method introduces coplanarity errors and formulates an objective function in the object space by employing orthogonal projections. It uses an alternative optimization strategy to fully exploit the line constraint information, leading to very accurate and efficient performance. For future work, we are interested in using our methods for real applications, for example, robot navigation.

This work was supported by the Natural Science Foundation of China (NSFC) under grant 11072263.

References

1. 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**, 381–395 (1981).
2. P. David, D. Dementhon, R. Duraiswami, and H. Samet, "SoftPOSIT: Simultaneous pose and correspondence determination," Int. J. Comput. Vis. **59**, 259–284 (2004).
3. P. David, D. Dementhon, R. Duraiswami, and H. Samet, "Simultaneous pose and correspondence determination using line features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003* (IEEE, 2003), pp. II-424–II-431.
4. F. Moreno-Noguer, V. Lepetit, and P. Fua, "Pose priors for simultaneously solving alignment and correspondence," in *Proceedings of the 10th European Conference on Computer Vision: Part II* (Springer-Verlag, 2008), pp. 405–418.
5. M. Dhome, M. Richetin, J. thierry Lapreste, and G. Rives, "Determination of the attitude of 3-D objects from a single perspective view," IEEE Trans. Pattern Anal. Mach. Intell. **11**, 1265–1278 (1989).
6. D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," Int. J. Comput. Vis. **15**, 123–141 (1995).
7. R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object pose: the link between weak perspective, paraperspective and full perspective," Int. J. Comput. Vis. **22**, 173–189 (1997).
8. S. Christy and R. Horaud, "Iterative pose computation from line correspondences," Comput. Vis. Image Understand. **73**, 137–144 (1999).
9. C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 610–622 (2000).
10. A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," IEEE Trans. Pattern Anal. Mach. Intell. **25**, 578–589 (2003).
11. A. Castro, Y. Frauel, E. Tepichin, and B. Javidi, "Pose estimation from a two-dimensional view by use of composite correlation filters and neural networks," Appl. Opt. **42**, 5882–5890 (2003).
12. F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative O(n) solution to the PnP problem," in *Proceedings of the IEEE 11th International Conference on Computer Vision, 2007* (IEEE, 2007), pp. 1–8.
13. W. E. L. Grimson, "Object recognition by computer: the role of geometric constraints," Int. J. Comput. Vis. **31**, 350–504 (1990).
14. J. S. Beis and D. G. Lowe, "Indexing without invariants in 3D object recognition," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 1000–1015 (1999).
15. Y. Lamdan and H. J. Wolfson, "Geometric hashing: a general and efficient model-based recognition scheme," in *Proceedings of the Second International Conference on Computer Vision* (IEEE, 1988), pp. 238–249.
16. P. Wunsch and G. Hirzinger, "Registration of cad models to images by iterative inverse perspective matching," in *Proceedings of the 13th International Conference on Pattern Recognition* (IEEE, 1996), pp. 78–83.
17. J. R. Beveridge and E. M. Riseman, "Optimal geometric model matching under full 3d perspective," Comput. Vis. Image Understand. **61**, 351–364 (1995).
18. F. Jurie, "Solution of the simultaneous pose and correspondence problem using gaussian error model," Comput. Vis. Image Understand. **73**, 357–373 (1999).
19. R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose estimation from corresponding point data," IEEE Trans. Syst. Man Cybern. **19**, 1426–1446 (1989).
20. R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object pose: the link between weak perspective, paraperspective, and full perspective," Int. J. Comput. Vis. **22**, 173–189 (1997).
21. L. Quan and Z. Lan, "Linear N-point camera pose determination," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 774–780 (1999).
22. H. H. Chen, "Pose determination from line-to-plane correspondences: existence condition and closed-form solutions," IEEE Trans. Pattern Anal. Mach. Intell. **13**, 530–541 (1991).
23. Y. Liu, T. S. Huang, and O. D. Faugeras, "A linear algorithm for motion estimation using straight line correspondences," Comput. Vis. Graph. Image Process. **44**, 35–57 (1988).
24. Y. Liu, T. S. Huang, and L. D. Faugeras, "Determination of camera location from 2-D to 3-D line and point correspondences," IEEE Trans. Pattern Anal. Mach. Intell. **12**, 28–37 (1990).
25. C.-N. Lee and R. M. Haralick, "Statistical estimation for exterior orientation from line-to-line correspondences," Image Vis. Comput. **14**, 379–388 (1996).
26. R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," Comput. Vis. Graph. Image Process. **60**, 313–342 (1994).
27. T. Q. Phong, R. Horaud, A. Yassine, and P. D. Tao, "Object pose from 2D to 3D point and line correspondences," Int. J. Comput. Vis. **15**, 225–243 (1995).
28. N. Navab and O. Faugeras, "Monocular pose determination from lines: critical sets and maximum number of solutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 1993), pp. 254–260.
29. S. Umeyama, "Least-squares estimation of transformatio parameters between two point patterns," IEEE Trans. Pattern Anal. Mach. Intell. **13**, 376–380 (1991).
30. B. K. P. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," J. Opt. Soc. Am. **5**, 1127–1135 (1988).
31. H. Wuest, F. Vial, and D. Stricker, "Adaptive line tracking with multiple hypotheses for augmented reality," in *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality* (IEEE, 2005), pp. 62–69.