ITERATIVE DETERMINATION OF CAMERA POSE FROM LINE FEATURES

Xiaohu Zhang, Xiangyi Sun, Yun Yuan, Zhaokun Zhu, Qifeng Yu

College of Aerospace and Materials Engineering, National University of Defense Technology
Changsha, 410073, P.R.China - zxh1302@hotmail.com
Hunan Key Laboratory of Videometrics and Vision Navigation, Changsha, 410073, P.R.China

Commission III/1

KEY WORDS: pose estimation, line feature, orthogonal iteration

ABSTRACT:

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. Experiments show that the algorithm has strong robustness to noise and outliers, and that it can attain very accurate results efficiently.

1 INTRODUCTION

Camera pose estimation is a basic task in photogrammetry and computer vision, and has many applications in visual navigation, object recognition, augmented reality, and *etc.*

The problem of pose estimation has been studied for a long time in the community of photogrammetry and computer vision, and numerous 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 (Haralick et al., 1989, Horaud et al., 1997, Quan and Lan, 1999, Moreno-Noguer et al., 2007).

Although the point feature is first used in pose estimation, line feature, which has the advantages of robust detection and having more structural information, is gaining increasing attentions. Typically, in the 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 features are less likely to be affected by occlusions thanks to multi-pixel support.

Closed-form algorithms were derived for three-line correspondences but multiple solutions may appear (Dhome et al., 1989, Chen, 1991). Linear solution(Ansar and Daniilidis, 2003) 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 (Dhome et al., 1989, Liu et al., 1988, Chen, 1991, Ansar and Daniilidis, 2003) can be used. In order to obtain more accurate results, iterative algorithms based on nonlinear optimization (Liu et al., 1990, Lee and Haralick, 1996, Christy and Horaud, 1999) are generally required. However, they generally do not fully exploit the specific structure of 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 Newton and Levenberg-Marquardt method may lack sufficient efficiency (Phong et al., 1995, Lu et al., 2000).

One interesting exception among the iterative algorithms is the Orthogonal Iteration(OI) algorithm developed for point features (Lu et al., 2000), which is not only accurate, but also robust to corrupted data and be 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 rotation matrices in a global convergent manner.

Inspired by this method, 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 which fully exploits the line constraints information can attain accurate and robust results efficiently even under strong noise and outliers.

2 CAMERA MODEL

The geometric model of a camera is depicted in Fig. 1. Let $c - x_c y_c z_c$ be the camera coordinate system with the origin fixed at the focal point, and the axis z_c coinciding with the optical axis and pointing to the front of the camera. I denotes the normalized 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 on the image plane. 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 (Dhome et al., 1989). In the object coordinate system, L_i can be described as $\lambda \mathbf{d}_i + \mathbf{P}_i$, where $\mathbf{d}_i = (d_i^x, d_i^y, d_i^z)^T$ is the unit direction of the line and, $\mathbf{P}_i = (x_i, y_i, z_i)^T$ is an arbitrary point on the line, and λ is a scalar. The 2D image line l_i in the camera coordinate system can be expressed as: $a_i x + b_i y + c_i = 0$. We define a unit vector $\mathbf{n}_i = (a_i, b_i, c_i)^T$, which represents l_i as $(x, y, 1) \cdot \mathbf{n}_i = 0$. It is clear that $\mathbf{n}_i$ is the normal vector of the interpretation plane.

The direction vector $\mathbf{d}_i$ and the point $\mathbf{P}_i$ can be expressed in the camera coordinate system as $\mathrm{R}\mathbf{d}_i$ and $\mathrm{R}\mathbf{P}_i + \mathbf{t}$, 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:

$$\mathbf{n}_i^T \mathrm{R}\mathbf{d}_i = 0, \tag{1}$$

$$\mathbf{n}_i^T (\mathrm{R}\mathbf{P}_i + \mathbf{t}) = 0. \tag{2}$$

Therefore each line correspondence provides two constraints. Equations (1) and (2) are the well known fundamental equations of the pose estimation problem from 2D to 3D line correspondences (Navab and Faugeras, 1993).

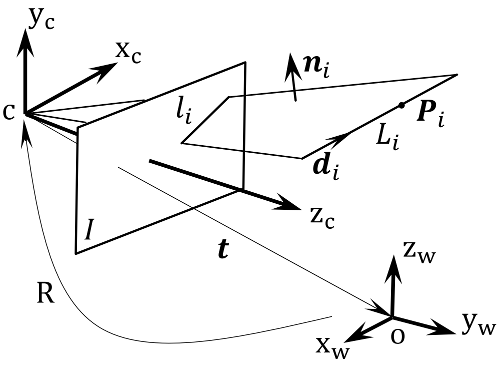


Figure 1: The geometry of the camera model

3 POSE FROM LINE CORRESPONDENCES

We developed a new algorithm of pose estimation from line correspondences. Our algorithm is inspired by Lu et al.'s method (Lu et al., 2000) which addresses pose estimation problem from points. In comparison with other algorithms, the iterative method of (Lu et al., 2000) can attain very accurate results in a fast and globally convergent manner. It is regarded as one of the most accurate and efficient pose estimation methods (Moreno-Noguer et al., 2007).

For pose estimation from line features, the existing linear algorithms (Ansar and Daniilidis, 2003, Liu et al., 1990) often lacks sufficient accuracy while the iterative algorithms (Phong et al., 1995, Kumar, 1994), which generally uses classical optimization techniques such as Newton and Levenberg-Marquardt method, may not fully exploit the specific structure of the pose estimation problem (Lu et al., 2000), and are usually sensitive to the initial noise or outliers. Our new iterative algorithm is an extension of the algorithm (Lu et al., 2000) to lines, which is named as Line based Orthogonal Iteration algorithm (LOI). Experiments in Sect. 4 demonstrate that our method is very efficient and can attain very good performances in terms of both accuracy and robustness.

3.1 Object-Space Objective Function

The fundamental equations indicate that each 3D-to-2D line correspondence provides 2 constraints. If N-line correspondences are available, the pose problem becomes the problem of minimizing the following objective function (Phong et al., 1995, Lee and Haralick, 1996, Kumar, 1994):

$$E(\mathrm{R},\mathbf{t}) = \sum_{i=1}^{N} (\mathbf{n}_i^T \mathrm{R}\mathbf{d}_i)^2 + \sum_{i=1}^{N} (\mathbf{n}_i^T (\mathrm{R}\mathbf{P}_i + \mathbf{t}))^2. \tag{3}$$

Instead of using the objective function of Eq. (3), we present and use an objective metric based on the coplanarity error in the object space. In fact, just as the collinearity of point correspondence in the way of orthogonal projection (Liu et al., 1990), the coplanarity of a 3D-to-2D line correspondence can be understood in the way that the projection of 3D line in the interpretation plane should be coincide with its self. In the camera coordinate system, the line direction vector is $\mathrm{R}\mathbf{d}_i$ and its projection in the interpretation plane can be obtained by $(\mathrm{I} - \mathbf{n}_i\mathbf{n}_i^T)\mathrm{R}\mathbf{d}_i$ where I is a 3 × 3 identity matrix. Let $\mathrm{K}_i = \mathrm{I} - \mathbf{n}_i\mathbf{n}_i^T$, we have:

$$\mathrm{R}\mathbf{d}_i = \mathrm{K}_i\mathrm{R}\mathbf{d}_i. \tag{4}$$

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

$$\mathrm{R}\mathbf{P}_i + \mathbf{t} = \mathrm{K}_i(\mathrm{R}\mathbf{P}_i + \mathbf{t}). \tag{5}$$

From the Eqs. (4) and (5), we have the corresponding vector differences as:

$$\mathbf{e}_i^{\mathbf{d}} = (\mathrm{I} - \mathrm{K}_i)\mathrm{R}\mathbf{d}_i, \tag{6}$$

$$\mathbf{e}_i^{\mathbf{P}} = (\mathrm{I} - \mathrm{K}_i)(\mathrm{R}\mathbf{P}_i + \mathbf{t}). \tag{7}$$

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

$$E_1(\mathrm{R}) = \sum_{i=1}^{N} \|\mathbf{e}_i^{\mathbf{d}}\|^2 = \sum_{i=1}^{N} \|(\mathrm{I} - \mathrm{K}_i)\mathrm{R}\mathbf{d}_i\|^2, \tag{8}$$

$$E_2(\mathrm{R},\mathbf{t}) = \sum_{i=1}^{N} \|\mathbf{e}_i^{\mathbf{P}}\|^2 = \sum_{i=1}^{N} \|(\mathrm{I} - \mathrm{K}_i)(\mathrm{R}\mathbf{P}_i + \mathbf{t})\|^2. \tag{9}$$

Compared with Eq. (3), the two objective functions (8) and (9) have clear geometry meaning that the optimal solution of pose should achieve the minimum value of the sum of squares of coplanarity errors (See Fig. 2).

Note that like the line-of-sight projection matrix V_i defined in (Liu et al., 1990), K_i is an interpretation plane projection matrix that, when applied to a object vector, projects the vector orthogonally to the interpretation plane. It owns the following properties:

$$\|\mathrm{K}_i\| \geq \|\mathrm{K}_i\mathbf{x}\|, \qquad \mathbf{x} \in \mathbb{R}^3, \tag{10}$$

$$\mathrm{K}_i^T = \mathrm{K}_i, \tag{11}$$

$$\mathrm{K}_i^2 = \mathrm{K}_i\mathrm{K}_i^T = \mathrm{K}_i. \tag{12}$$

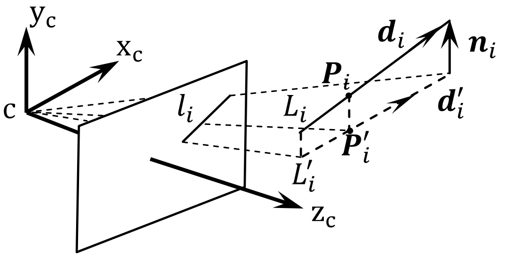


Figure 2: Object-Space error of line correspondence

Since our new pose estimation method involves mainly the least-squares estimation, we show a lemma given by the work of (Umeyama, 1991) here before proceeding to the details of our method.

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

$$f(\mathrm{R}) = \|\mathrm{A} - \mathrm{RB}\|^2, \tag{13}$$

the optimal rotation matrix R such that

$$\mathrm{R} = \mathrm{USV}^T. \tag{14}$$

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

$$\mathrm{S} = \begin{cases} \mathrm{I} & if\ det(\mathrm{AB}^T) \geq 0, \\ diag(1,1,\cdots,1,-1) & else. \end{cases} \tag{15}$$

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

$$\mathrm{S} = \begin{cases} \mathrm{I} & if\ det(\mathrm{U})det(\mathrm{V}) = 1, \\ diag(1,1,\cdots,1,-1) & if\ det(\mathrm{U})det(\mathrm{V}) = -1. \end{cases} \tag{16}$$

Algorithm 1: Rotation then Translation

```
1. Given $N$ ($N \geq 3$) 3D-to-2D line correspondences and initial
   rotation $\mathrm{R}_0$, compute $\mathrm{K}_i$ for $i = 1, \dots, N$. Define
   $\mathrm{B} = (\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N)$. Set $k = 0$.
2. Perform the following steps:
   (a) Compute $\mathrm{A} = (\mathrm{K}_1\mathrm{R}_k\mathbf{d}_1, \cdots, \mathrm{K}_N\mathrm{R}_k\mathbf{d}_N)$.
   (b) Compute $\mathrm{M} = \mathrm{AB}^T$ and perform SVD: $\mathrm{UDV}^T = \mathrm{M}$.
   (c) Compute $\mathrm{R}_{k+1} = \mathrm{USV}^T$, where S is set according to
       Eqs. (15) and (16).
   (d) Terminate the iteration if convergence is reached;
       otherwise, $k = k + 1$, go to setp (a).
3. Compute translation $\mathbf{t}$ via (18).
```

3.2 LOI-1: Rotation then Translation

We propose a pose estimation algorithm which optimizes alternatively on the rotation matrix and translation vector.

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:

$$\mathrm{A} = (\mathrm{K}_1\mathrm{R}\mathbf{d}_1, \mathrm{K}_2\mathrm{R}\mathbf{d}_2, \cdots, \mathrm{K}_N\mathrm{R}\mathbf{d}_N),$$

$$\mathrm{B} = (\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N).$$

$E_1(\mathrm{R})$ then becomes :

$$E_1(\mathrm{R}) = \|\mathrm{A} - \mathrm{RB}\|^2. \tag{17}$$

It is seen that Eq. (17) bears a close resemblance to Eq. (13). This naturally lets us compute R iteratively as follows: given the estimation rotation matrix R_k at k-th iteration, we compute $\mathrm{A}(\mathrm{R}_k)$ and seek to minimize $\|\mathrm{A}(\mathrm{R}_k) - \mathrm{RB}\|^2$ to get the next estimate R_{k+1} according to the Lemma (1). This step is repeated until convergence is achieved.

After attaining an estimate of rotation R, the optimal estimate of translation $\mathbf{t}$ can be computed easily by minimizing $E_2(\mathrm{R}, \mathbf{t})$ of Eq. (9):

$$\mathbf{t} = t(\mathrm{R}) = (\sum_{i=1}^{N}(\mathrm{I} - \mathrm{K}_i))^{-1} \sum_{i=1}^{N}(\mathrm{K}_i - \mathrm{I})\mathrm{R}\mathbf{P}_i. \tag{18}$$

Clearly the matrix $(\sum_{i=1}^{N}(\mathrm{I} - \mathrm{K}_i))$ must be invertible for Eq. (18) to hold. Since $\mathrm{I} - \mathrm{K}_i = \mathbf{n}_i\mathbf{n}_i^T$, we have $(\sum_{i=1}^{N}(\mathrm{I} - \mathrm{K}_i)) = \mathrm{CC}^T$, where

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

Hence if $rank(\mathrm{CC}^T) \equiv 3$, Eq. (18) can be well-defined. In fact, $rank(\mathrm{CC}^T) = rank(\mathrm{C}) = 3$ 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. (18).

We now achieve a two-step pose estimation algorithm(LOI-1): firstly the optimal R^* is iteratively computed, and then the best translation $\mathbf{t}^*$ is obtained given the estimated R^*. The algorithm is summarized in Algorithm 1.

3.3 LOI-2: Alternative Optimization

In Algorithm 1, the objective function $E_1(R)$ is minimized firstly to solve for rotation R. The estimate is then used to minimize the objective function $E_2(\mathrm{R}, \mathbf{t})$ to determine the translation $\mathbf{t}$. This method only uses the information of line direction to compute rotation, and doesn't use the set of constraints effectively. In the framework of solving rotation and translation separately, the small errors in the rotation estimation are amplified into large errors in the translation stage (Kumar, 1994). To fully exploit the set of constraints, we can modify Algorithm 1 to optimize alternatively on the rotation matrix and translation vector.

Assuming we have obtained the k-th estimation of R_k, $\mathbf{t}_k$ is computed via $t(\mathrm{R}_k)$. We firstly use the rotation estimation iterative step of Algorithm 1 to estimate a new rotation value $\mathrm{R'}_{k+1}$, and obtain a new estimate of translation $\mathbf{t'}_{k+1}$ via $t(\mathrm{R'}_{k+1})$ from Eq. (18). Then with $(\mathrm{R'}_{k+1}, \mathbf{t'}_{k+1})$, we use the method of (Lu et al., 2000) to obtain the final $(k+1)$-th estimation by minimizing the objective function $E_2(\mathrm{R}, \mathbf{t})$. The last step is described as follows.

In the algorithm of (Lu et al., 2000), R and $\mathbf{t}$ are iteratively optimized by minimizing an object space objective function defined for the point correspondences:

$$E(\mathrm{R}, \mathbf{t}) = \sum_{i}^{N} \|(\mathrm{I} - \mathrm{V}_i)(\mathrm{R}\mathbf{P}_i + \mathbf{t})\|^2, \tag{19}$$

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:

$$\mathrm{R}^{k+1} = \arg\min_{\mathrm{R}} \sum_{i=1}^{N} \|\mathrm{R}\mathbf{P}_i + \mathbf{t} - \mathrm{V}_i\mathbf{q}_i^k\|^2, \tag{20}$$

where $\mathbf{q}_i^k = \mathrm{R}^k\mathbf{P}_i + \mathbf{t}^k$. This absolute orientation problem is then solved by *SVD* method (Horn et al., 1988).

It is seen that the only difference in Eq. (9) compared with Eq. (19) is the use of projection matrix K_i instead of V_i. Both projection vectors K_i and V_i bear the same properties (Sect. 3.1). Hence after we have obtained the estimates $(\mathrm{R'}_{k+1}, \mathbf{t'}_{k+1})$, an estimate of rotation, R_{k+1}, can be computed by directly using the algorithm of (Lu et al., 2000) to minimize the objective function (9). The new algorithm is summarized in Algorithm 2.

Obviously if given an initial estimate for both R and $\mathbf{t}$, purely minimizing $E_2(\mathrm{R}, \mathbf{t})$ by using the the method of (Lu et al., 2000) leads to another algorithm (we denote it as LOI-3). Since the only difference is the definition and use of a projection vector, we will not go further on this algorithm. For more details, the readers are referred to (Lu et al., 2000).

4 EXPERIMENTS

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

4.1 Synthetic Data

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. **??**) 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.

Algorithm 2: Alternative Optimization

```
1. Given $N$ ($N \geq 3$) 3D-to-2D line correspondences and initial
   rotation $\mathrm{R}_0$, compute: $\mathrm{K}_i$ for $i = 1, \ldots, N$,
   $B = (\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N)$. Set $k = 0$.
2. Perform the following steps:
   (a) Compute $\mathrm{A} = (\mathrm{K}_1\mathrm{R}_k\mathbf{d}_1, \cdots, \mathrm{K}_N\mathrm{R}_k\mathbf{d}_N)$.
   (b) Compute $\mathrm{M} = \mathrm{AB}^T$ and perform $SVD$: $\mathrm{UDV}^T = \mathrm{M}$.
   (c) Compute $\mathrm{R}'_{k+1} = \mathrm{USV}^T$, where S is set according to
       Eqs. (15) and (16).
   (d) Compute $\mathbf{t}'_{k+1} = t(\mathrm{R}'_{k+1})$
   (e) Given $\mathrm{R}'_{k+1}$ and $\mathbf{t}'_{k+1}$, compute $\mathrm{R}_{k+1}$ using the
       algorithm of (Lu et al., 2000).
   (f) Compute $\mathbf{t}_{k+1} = t(\mathrm{R}_{k+1})$.
   (g) Terminate the iteration if convergence is attained;
       otherwise, $k = k + 1$, go to step (a).
```

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. To facilitate the description, we denote the Algorithm 1 $\sim$ 3 as LOI-1, LOI-2 and LOI-3 separately.

In Fig. 3, we plot the rotation and translation relative errors produced by the three algorithms as a function of Gaussian noise with its standard deviation varies 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-2” is consistently more accurate than the other two algorithms.

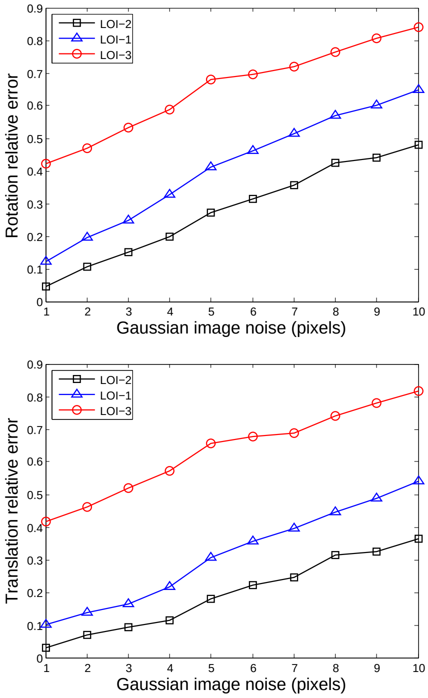


Figure 3: Relative rotation and translation error as a function of image noise when the number of lines is fixed to be 8.

Fig. 4 plots the errors as the function of the number of 3D object lines when the image noise is fixed ($\sigma = 3$ pixels). We compare the performances of our algorithms with the OI method (Lu et al., 2000), for which the corresponding $2N$ endpoints of the 3D lines are used. It can be seen that all these algorithms can achieve higher accuracy when the number of feature correspondence increases. LOI-2 and OI algorithms show more accurate and stable performances.

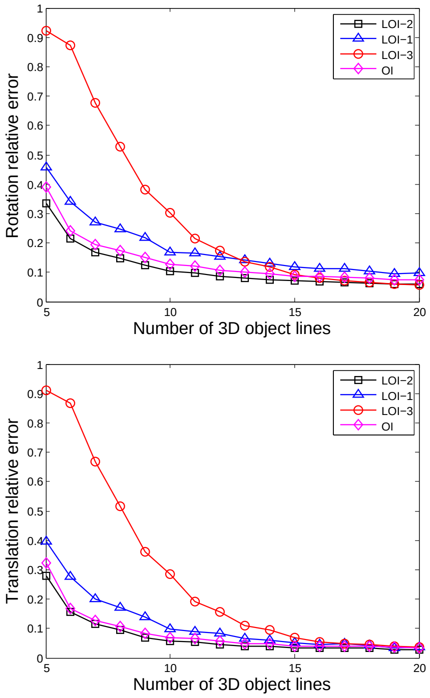
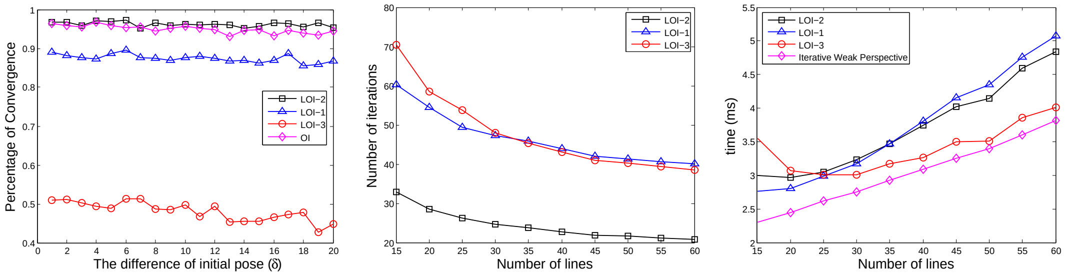


Figure 4: Relative rotation and translation error as a function of the number of object lines when the standard deviation of image noise is fixed to be 3 pixels.

In Fig. 5(a), we give the percentage of convergence when the initial poses are generated from a multinormal distribution with mean as the true pose and the diagonal covariance $\delta\Sigma_0$, where the standard deviation element of Σ_0 is about 1.5 degree 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 LOI-2 shows very robust performance and slightly outperforms the OI algorithm which is proved to be global convergent. In contrast, the LOI-3 algorithm produces very poor performance. We conclude that, without exploiting the direction information of the lines, LOI-3 is very sensitive to the image noise as well as to the initial pose. Fig. 5(b) plots the number of iterations as the function of the number of object lines. With the increase of the line number, the number of iterations needed decreases. Since in LOI-2 there are two updates of rotation, the computation time taken by 1 iteration in LOI-2 is about double of that in LOI-1 and LOI-3. This can be seen from Fig. 5(c), which gives the computation times. The LOI-1 and LOI-2 use almost the same running times. LOI-3 is faster with the increasing of the number of lines. We compare our methods with the iterative weak perspective (IWP) method (Christy and Horaud, 1999), which estimates a pose with a weak perspective camera model and improves the estimation iteratively by solving an approximate system of linear equations. Our orthogonal iteration methods are very efficient and comparable to the IWP method.

4.2 Real Data

We also validated our pose estimation approach for line correspondences, by using the algorithm for 3D line object tracking.



(a) (b) (c)

Figure 5: (a) the number of iterations and (b) the running times as a function of the number of lines. (c) the percentage of convergence when initial poses are generated from a multinormal distribution.

The tracking is performed on an image sequence recorded from a moving calibrated camera pointing towards the scene as shown in Fig. 6. 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 gradient maxima for a set of sampled points in the line (Wuest et al., 2005). The strong maxima are taken as the 2D feature points whose corresponding 3D sampled points are 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. Robust pose estimation method that well resists outliers is evaluable for robust tracking. We use our method, LOI-2 specifically, for the tracking. Fig. 6 shows four frames of the tracked sequence. Our method consistantly tracks the whole sequence.

5 CONCLUSIONS AND FUTURE WORK

Robust pose estimation is necessary for refining the pose. We presented efficient and robust iterative pose estimation algorithms for line features. Our method introduces coplanarity errors and formulates objective function in the object space by employing orthogonal projections. In the same framework, three pose estimation algorithms are given and their performances are evaluated. Compared with other pose estimation algorithm for lines, one of the proposed methods-LOI-2 algorithm is extremely robust, accurate, and also converges fast.

For future work, we are interested in using our methods for real applications, for example, robot navigation. By making use of more other information like the appearance of object, the search of pose from unknown line correspondences may speed up. We are also interested in implementing our simultaneous pose and correspondence method on GPU, for real-time virtual reality applications.

REFERENCES

Ansar, A. and Daniilidis, K., 2003. Linear pose estimation from points or lines. IEEE transactions on pattern analysis and machine intelligence.

Chen, H. H., 1991. Pose determination from line-to-plane correspondences: Existence condition and closed-form solutions. IEEE transactions on pattern analysis and machine intelligence.

Christy, S. and Horaud, R., 1999. Iterative pose computation from line correspondences. Computer vision and image understanding.

Dhome, M., Richetin, M., thierry Lapreste, J. and Rives, G., 1989. Determination of the attitude of 3-D objects from a single perspective view. IEEE transactions on pattern analysis and machine intelligence.

Haralick, R. M., Joo, H., Lee, C.-N., Zhuang, X., VAIDYA, V. G. and Kim, M. B., 1989. Pose estimation from corresponding point data. International Journal of Computer Vision.

Horaud, R., dornaika, F., Lamiroy, B. and christy, S., 1997. Object pose: the link between weak perspective, paraperspective, and full perspective. International Journal of Computer Vision 22(2), pp. 173–189.

Horn, B. K. P., Hilden, H. and Negahdaripour, S., 1988. Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society America 5(7), pp. 1127–1135.

Kumar, P., 1994. Robust methods for estimating pose and a sensitivity analysis. Image understanding 60(3), pp. 313–342.

Lee, C.-N. and Haralick, R. M., 1996. Statistical estimation for exterior orientation from line-to-line correspondences. Image and vision computing 14, pp. 379–388.

Liu, Y., Huang, T. S. and Faugeras, L. D., 1990. Determination of camera location from 2-D to 3-D line and point correspondences. IEEE transactions on pattern analysis and machine intelligence.

Liu, Y., Huang, T. S. and Faugeras, O. D., 1988. A linear algorithm for motion estimation using straight line correspondences. Computer Vision Graphics Image Processing.

Lu, C.-P., Hager, G. D. and Mjolsness, E., 2000. Fast and globally convergent pose estimation from video images. IEEE transactions on pattern analysis and machine intelligence.

Moreno-Noguer, F., Lepetit, V. and Fua, P., 2007. Accurate non-iterative O(n) solution to the PnP problem. In: IEEE Conference on Computer Vision and Pattern Recognition.

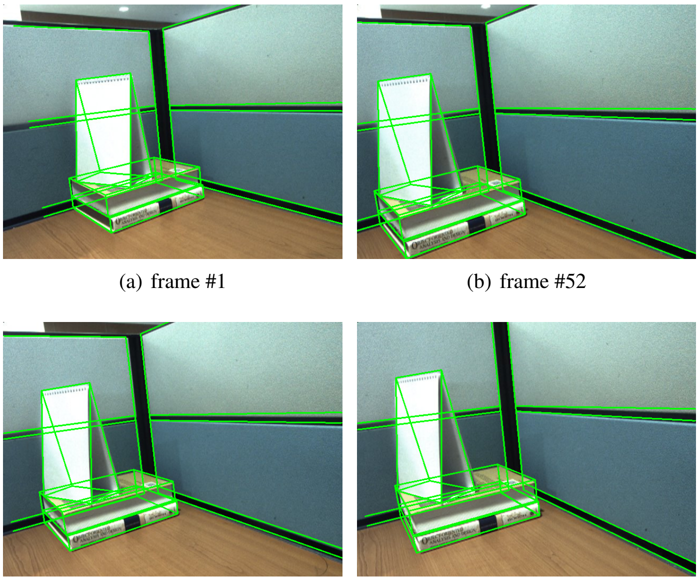
Navab, N. and Faugeras, O., 1993. Monocular pose determination from lines: Critical sets and maximum number of solutions. In: IEEE Conference on Computer Vision and Pattern Recognition.

Phong, T. Q., Horaud, R., Yassine, A. and Tao, P. D., 1995. Object pose from 2d to 3d point and line correspondences. International Journal of Computer Vision 15(3), pp. 225–243.

Quan, L. and Lan, Z., 1999. Linear N-point camera pose determination. IEEE transactions on pattern analysis and machine intelligence.

Umeyama, S., 1991. Least-squares estimation of transformatio parameters between two point patterns. IEEE transactions on pattern analysis and machine intelligence.

Wuest, H., Vial, F. and Stricker, D., 2005. Adaptive line tracking with multiple hypotheses for augmented reality. In: 4th IEEE and ACM International Symposium on Mixed and Augmented Reality.



(a) frame #1

(b) frame #52

(c) frame #86

(d) frame #135

Figure 6: Pose estimation for 3D object tracking on a sequnce recorded with a moving camera.