Pose Estimation using Both Points and Lines for Geo-Localization

Srikumar Ramalingam[1] Sofien Bouaziz[2] Peter Sturm[3]
[1]Mitsubishi Electric Research Lab (MERL), Cambridge, MA, USA
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[3]INRIA Grenoble – Rhône-Alpes and Laboratoire Jean Kuntzmann, Grenoble, France
{srikumar.ramalingam}@merl.com, {sofien.bouaziz}@gmail.com {peter.sturm}@inrialpes.fr

Abstract— **This paper identifies and fills the probably last two missing items in minimal pose estimation algorithms using points and lines. Pose estimation refers to the problem of recovering the pose of a calibrated camera given known features (points or lines) in the world and their projections on the image. There are four minimal configurations using point and line features: 3 points, 2 points and 1 line, 1 point and 2 lines, 3 lines. The first and the last scenarios that depend solely on either points or lines have been studied a few decades earlier. However the mixed scenarios, which are more common in practice, have not been solved yet. In this paper we show that it is indeed possible to develop a general technique that can solve all four scenarios using the same approach and that the solutions involve computing the roots of either a 4th degree or an 8th degree equation. The centerpiece of our method is a simple and generic method that uses collinearity and coplanarity constraints for solving the pose. In addition to validating the performance of these algorithms in simulations, we also show a compelling application for geo-localization using image sequences and coarse (plane-based) 3D models of GPS-challenged urban canyons.**

I. Introduction and Previous Work

In robotics and vision community, several promising simultaneous localization and mapping (SLAM) algorithms have been developed in the last three decades and detailed surveys are available [5]. Existing techniques in SLAM can be classified into ones that use a motion model [2], [3] and the approaches free of motion models [21], [27]. The basic idea in using a motion model is to smooth the trajectory of the camera and constrain the search area for feature correspondences. On the other hand, the ones without using a motion model reconstruct the scene coarsely using 3D reconstruction algorithms and estimate the pose of the camera w.r.t the coarse model. In contrast to many methods where both the 3D reconstruction and localization are solved simultaneously or sequentially, our method attempts to solve only the localization problem assuming that a coarse 3D model of the city is already given.

Recent years in computer vision have seen a wide variety of geometrical problems being addressed for cases of minimal amounts of image features. The classical approach is to use all the available features and solve it using some least squares measure over all features. However, in many vision problems minimal solutions have proven to be less noise-prone compared to non-minimal algorithms: they have been very useful in practice as hypothesis generators in hypothesize-and-test algorithms such as RANSAC [7]. Minimal solutions have been proposed for several computer vision problems: auto-calibration of radial distortion [16], perspective three point problem [8], the five point relative pose problem [19], the six point focal length problem [29], the six point generalized camera problem [30], the nine point problem for estimating para-catadioptric fundamental matrices [9] and the nine point radial distortion problem [18]. The last few years have seen the use of minimal problems in various applications [28] and there are even unification efforts to keep track of all the existing solutions[1].

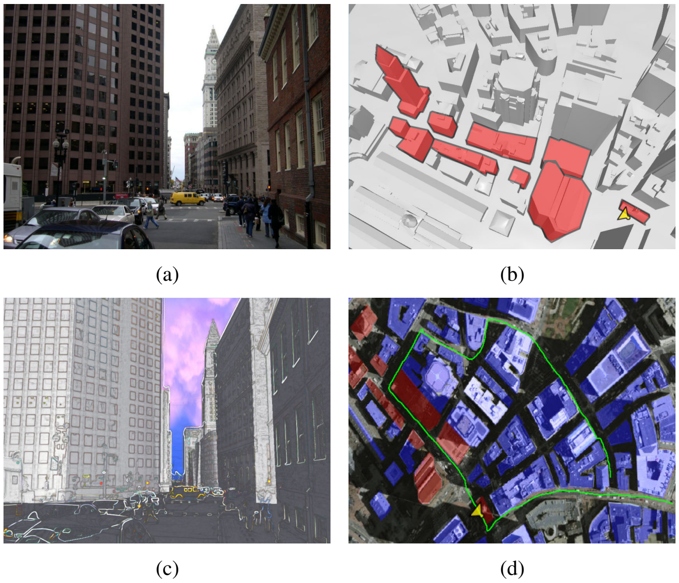


Fig. 1: *Geo-localization using points and lines. (a) Real image. (b) The buildings visible in the real image are marked in the 3D model of the city. (c) Reprojection of the edges from real image on the 3D model after geo-localization. (d) Location of the image shown in (a) computed using our algorithm.*

a) 2D-3D Registration: In this work we revisit one of the very old problems in computer vision and robotics: pose estimation using points and lines. Given three correspondences between points/lines in the world and their projections on the images, the goal is to compute the pose of the camera in the world coordinate system. The solution for three lines was given by Dhome et al. [4]. The solution to three points case was given even before - Grunert [10], Fischler and Bolles [7], Church's method [6], Haralick et al. [11], to

[1] http://cmp.felk.cvut.cz/minimal/

name but a few references. To the best of our knowledge, we are not aware of any minimal solution for the mixed scenarios. However, in practice both point and line features have complementary advantages. Although, the fusion of points and lines for tracking has been studied in the past, minimal solutions which are useful to achieve robustness to outliers, insufficient correspondences and narrow fields of view have not been considered. In this work we propose a pose estimation solution using three features – it could be points, lines or both. There are several registration algorithms for 3D-3D scenarios though; for example [22], [13], [25]. A review of camera pose and relative motion estimation algorithms for non-central and other generalized camera models can be found in [31].

Our contribution is important because it is not always possible to obtain even three correct and non-degenerate line or point correspondences in real applications, both indoor and outdoor. As image-based localization is getting considerable attention in the recent years, we believe that this contribution is timely and will enable such applications in practice.

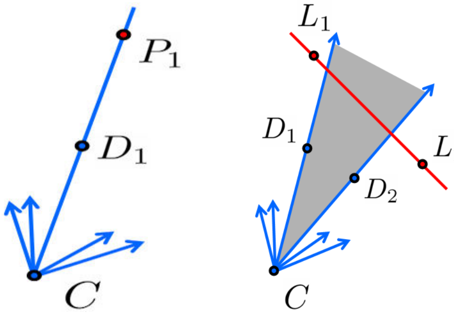
b) Image-based geo-localization: In the last few years, there has been an increasing interest in inferring geolocation from images [26], [35], [33], [14], [12], [23]. In [26], Robertson and Cipolla showed that it is possible to obtain geospatial localization by matching a query image with an image database using vanishing lines. Zhang and Kosecka showed accurate results in the ICCV 2005 computer vision contest (”Where am I?”) using SIFT features [35]. Jacobs et al. used a novel approach to geolocate a webcam by correlating its images with satellite weather imagery at the same time [14]. Hays and Efros used millions of GPS-tagged images from the web for georeferencing a new image [12]. In contrast to most of these approaches that leverage on the availability of these georeferenced images, we use coarse 3D models from the web for geospatial localization: like georeferenced images, a large repository of coarse 3D models already exists for major cities in the world. Koch and Teller proposed a localization method using a known 3D model and a wide angle camera for indoor scenes by matching lines from the 3D model with the lines in images [15]. In contrast to their work, our work relies only on minimal solutions and uses both points and lines for geolocalization. In our prior work, we show that skylines from omni-images are very unique and can serve as fingerprints for specific locations [23]. It is important to notice that skylines are nothing but piecewise-linear segments, consisting of points and lines, that separates buildings and sky. Accordingly, the skyline matching for geo-localization can be seen as a special case of the proposed algorithm.

c) Our contributions:

- Our first and main contribution is a general framework to solve all four minimal problems using two geometrical constraints: collinearity and coplanarity.
- Our second contribution is the use of intermediate coordinate frames for simplifying the equations involved in the pose estimation. A direct application of the constraints would lead to the solution of a 64th degree polynomial and up to 64 solutions. On the other hand, our choice of coordinate frames reduces this to 4th and 8th degree equations.
- We show promising results for geo-localization using coarse 3D models and image sequences (not videos).

II. Overview of our approach

A. Collinearity and Coplanarity



(a) (b)

Fig. 2: *(a) The minimal solutions proposed in this paper essentially use two geometric constraints: collinearity and coplanarity. In (a) the projection ray CD_1, linked to a 2D feature point, and the associated 3D scene point P_1 are collinear if expressed in the same reference frame. In (b), two projection rays CD_1 and CD_2, linked to the end points of a 2D line segment, and the associated 3D line represented by two points L_1 and L_2 are all coplanar.*

Our framework can solve all four minimal cases using only two geometric constraints: *collinearity* and *coplanarity*. The collinearity constraint comes from 2D-3D point correspondences. We use a generic imaging setup [24], every pixel in the image corresponds to a 3D projection ray. For example in Figure 2(a), we show a projection ray CD_1 and a scene point P_1 lying on it, if expressed in the same reference frame. Our goal is to find the pose $(\mathsf{R},\mathbf{T})$ under which the scene point P_1 lies on the ray CD_1. We stack these points in the following matrix, which we refer to as the *collinearity matrix*:

$$\begin{pmatrix} C_x & D_{1x} & R_{11}P_{1x}+R_{12}P_{1y}+R_{13}P_{1z}+T_1 \\ C_y & D_{1y} & R_{21}P_{1x}+R_{22}P_{1y}+R_{23}P_{1z}+T_2 \\ C_z & D_{1z} & R_{31}P_{1x}+R_{32}P_{1y}+R_{33}P_{1z}+T_3 \\ 1 & 1 & 1 \end{pmatrix} \quad (1)$$

The collinearity constraint will force the determinant of any 3×3 submatrix of the above matrix to vanish. In other words, we obtain four constraints by removing one row at a time. Although four equations arise from the above matrix, only two are independent and thus useful.

The second geometric constraint comes from 2D-3D line correspondences. As shown in Figure 2(b), the points C, D_1, D_2, L_1 and L_2 lie on a single plane if expressed in the same reference frame. In other words, for the correct pose $[\mathsf{R},\mathbf{T}]$ we obtain two constraints from a single 2D-3D line correspondence: the quadruplets $(C,D_1,D_2,[\mathsf{R},\mathbf{T}]L_1)$ and $(C,\ D_1,\ D_2,[\mathsf{R},\mathbf{T}]L_2)$ are each coplanar. The coplanarity condition for the quadruplet $(C,D_1,D_2,[\mathsf{R},\mathbf{T}]L_1)$ forces the determinant of the following matrix to vanish:

$$\begin{pmatrix} C_x & D_{1x} & D_{2x} & R_{11}L_{1x}+R_{12}L_{1y}+R_{13}L_{1z}+T_1 \\ C_y & D_{1y} & D_{2y} & R_{21}L_{1x}+R_{22}L_{1y}+R_{23}L_{1z}+T_2 \\ C_z & D_{1z} & D_{2z} & R_{31}L_{1x}+R_{32}L_{1y}+R_{33}L_{1z}+T_3 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{2}$$

Similarly the other quadruplet $(C,\ D_1,\ D_2,[\mathsf{R},\mathbf{T}]L_2)$ also gives a coplanarity constraint. Accordingly, every 2D-3D line correspondence gives 2 equations from the two points on the line.

Our goal is to compute 6 parameters (3 for R and 3 for $\mathbf{T}$) for which the 3D features (both points and lines) satisfy the collinearity and coplanarity constraints. Thus we have four possible minimal cases (3 points, 2 points and 1 line, 1 point and 2 lines, 3 lines).

B. The choice of reference frames

As shown in Figures 3 and 4 let us assume that the original camera and world reference frames, where the points and lines reside, are denoted by $\mathscr{C}_0$ and $\mathscr{W}_0$ respectively. Our goal is to compute the transformation $(\mathsf{R}_{w2c},\mathbf{T}_{w2c})$ which expresses the 3D points and lines in the camera reference frame. A straight-forward application of collinearity and coplanarity constraints will result in 6 linear equations involving 12 variables (9 R_{ij}'s, 3 T_i's). In order to solve these variables we need additional equations: these can be 6 quadratic orthogonality constraints on R_{w2c}. Methods for computing a polynomial solution need not result in a polynomial of the smallest possible degree. The solution of such a system will eventually result in a 64th degree polynomial equation. This may have up to 64 solutions (upper bound as per Bezout's theorem) and the computation of such solutions may not be feasible for several robotics applications.

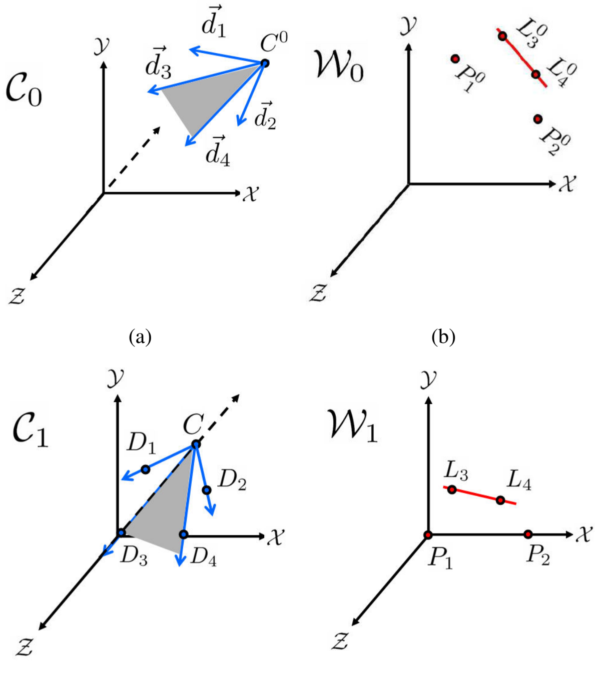
We provide a method to overcome this difficulty. In order to do this, we first transform both the camera and world reference frames $\mathscr{C}_0$ and $\mathscr{W}_0$ to $\mathscr{C}_1$ and $\mathscr{W}_1$ respectively. After this transformation our goal is to find the pose $(\mathsf{R},\mathbf{T})$ between these intermediate reference frames. We choose these reference frames $\mathscr{C}_1$ and $\mathscr{W}_1$ such that the resulting polynomial equation is of lowest possible degree. Our choice of coordinate frames reduces to 4th and 8th degree equations for the two mixed scenarios. Although we do not theoretically prove that our solutions are of the lowest possible degrees, we believe so because of the following argument. The best existing solutions for pose estimation using three points and three lines use 4th and 8th degree solutions respectively. Since mixed cases are in the middle, our solutions for (2 points, 1 line) and (1 point, 2 lines) cases use 4th and 8th degree solutions respectively. Recently, it was shown using Galois theory that the solutions that use the lowest possible degrees are the optimal ones [20].

In what follows we present pose estimation algorithms for the two minimal mixed cases.

III. Minimal Solutions

A. 2 Points and 1 Line

In this section, we provide a pose estimation algorithm from two 2D-3D point and one 2D-3D line correspondences. From the 2D coordinates of the points we can compute the corresponding projection rays using calibration. In the case of 2D lines, we can compute the corresponding projection rays for the end points of the line segment in the image. In what follows, we only consider the associated 3D projection rays for point and line features on the images.



(a) (b)

(c) (d)

Fig. 3: *The choice of intermediate reference frames $\mathscr{C}_1$ and $\mathscr{W}_1$ in the pose estimation for the two points plus one line case. The camera reference frames before and after the transformation are shown in (a) and (c) respectively. Similarly the world reference frames before and after the transformation are shown in (b) and (d) respectively. See text for details on these transformations.*

d) The choice of camera reference frame $\mathscr{C}_1$: In figure 3(a) and (b), we show the camera projection rays (associated with 2D points and lines) and 3D features (points and lines) in $\mathscr{C}_0$ and $\mathscr{W}_0$ respectively. In $\mathscr{C}_0$, let the center of the camera be C^0, the projection rays corresponding to the two 2D points be given by their direction vectors $\vec{d}_1$ and $\vec{d}_2$, the projection rays corresponding to the 2D line be given by direction vectors $\vec{d}_3$ and $\vec{d}_4$.

In the intermediate camera frame $\mathscr{C}_1$ we always represent the projection rays of the camera using two points (center and a point on the ray). Let the projection rays corresponding to the two 2D points be given by CD_1 and CD_2 and the line be given by CD_3 and CD_4. Let the plane formed by the triplet (C,D_3,D_4) be referred to as the *interpretation* plane. We choose an intermediate frame of reference $\mathscr{C}_1$ that satisfies the following conditions:

- The camera center is at $C(0,0,-1)$.
- One of the projection rays CD_3 corresponding to the line L_3L_4 is on the $\mathscr{Z}$ axis such that $D_3 = (0,0,0)$.
- The other projection ray CD_4 corresponding to the line L_3L_4 lies on the $\mathscr{X}\mathscr{Z}$ plane such that D_4 is on the $\mathscr{X}$ axis.

Now we show that such a transformation is possible for any set of projection rays corresponding to two points and one line using a constructive argument. Let P^0 and P denote the coordinates of any point in the reference frames $\mathscr{C}_0$ and $\mathscr{C}_1$ respectively. Following this notation, the points D_3 and D_4 are expressed in $\mathscr{C}_0$ and $\mathscr{C}_1$ using simple algebraic derivation:

$$\begin{aligned}
D_3^0 &= C^0 + \vec{d}_1, \\
D_4^0 &= C^0 + \frac{\vec{d}_2}{\vec{d}_1.\vec{d}_2}, \\
D_3 &= \mathbf{0}_{3\times 1}, \\
D_4 &= \begin{pmatrix} \tan(\cos^{-1}(\vec{d}_1.\vec{d}_2)) \\ 0 \\ 0 \end{pmatrix}
\end{aligned}$$

The pose $(\mathsf{R}_{c1}, \mathbf{T}_{c1})$ between $\mathscr{C}_0$ and $\mathscr{C}_1$ is given by the one that transforms the triplet (C^0, D_3^0, D_4^0) to (C, D_3, D_4).

e) The choice of world reference frame $\mathscr{W}_1$: Now we describe the choice of the intermediate world reference frame. Let the Euclidean distance between any two 3D points P and Q be denoted by $d(P,Q)$. The two 3D points and one 3D point on the 3D line in $\mathscr{W}_1$ are given below:

$$P_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, P_2 = \begin{pmatrix} d(P_1^0, P_2^0) \\ 0 \\ 0 \end{pmatrix}, L_3 = \begin{pmatrix} X_3 \\ Y_3 \\ 0 \end{pmatrix} \qquad (3)$$

where X_3 and Y_3 can be computed using simple trigonometry.

$$\begin{aligned}
X_3 &= (L_3 - P_1).\frac{(P_2 - P_1)}{d(P_1,P_2)}, \\
Y_3 &= d(L_3, P_1 + X_3.\frac{(P_2 - P_1)}{d(P_1,P_2)})
\end{aligned}$$

The pose $(\mathsf{R}_{w1}, \mathbf{T}_{w1})$ between $\mathscr{W}_0$ and $\mathscr{W}_1$ is given by the one that transforms the triplet (P_1^0, P_2^0, L_3^0) to (P_1, P_2, L_3).

For brevity, we use the following notation in the pose estimation algorithm.

$$D_{i=\{1,2\}} = \begin{pmatrix} a_i \\ b_i \\ 0 \end{pmatrix}, D_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} D_4 = \begin{pmatrix} a_4 \\ 0 \\ 0 \end{pmatrix}$$

$$P_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} P_2 = \begin{pmatrix} X_2 \\ 0 \\ 0 \end{pmatrix}, L_3 = \begin{pmatrix} X_3 \\ Y_3 \\ 0 \end{pmatrix}, L_4 = \begin{pmatrix} X_4 \\ Y_4 \\ Z_4 \end{pmatrix}$$

f) Pose estimation between $\mathscr{C}_1$ and $\mathscr{W}_1$: The first step is to stack all the available collinearity and coplanarity constraints. In this case we have two collinearity matrices for the triplets (C,D_1,P_1) and (C,D_2,P_2) corresponding to the 3D points P_1 and P_2 respectively. As shown in Equation (1), these two collinearity matrices give four equations. In addition, we have two coplanarity equations from the quadruplets (C,D_3,D_4,L_3) and (C,D_3,D_4,L_4) corresponding to the 3D line L_3L_4. On stacking the constraints from the determinants of (sub)-matrices we obtain the linear system $\mathscr{A}\mathscr{X} = \mathscr{B}$ where $\mathscr{A}$, $\mathscr{X}$ and $\mathscr{B}$ are given below:

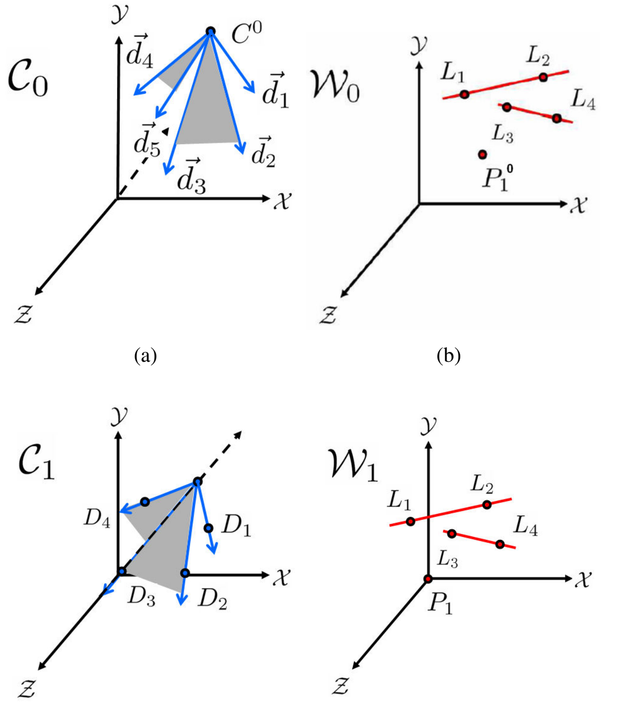
$$\mathscr{A} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & -b_1 & a_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & b_1 \\
-b_2X_2 & a_2X_2 & 0 & 0 & 0 & -b_2 & a_2 & 0 \\
0 & -X_2 & b_2X_2 & 0 & 0 & 0 & -1 & b_2 \\
0 & X_3 & 0 & Y_3 & 0 & 0 & 1 & 0 \\
0 & X_4 & 0 & Y_4 & Z_4 & 0 & 1 & 0
\end{pmatrix} \qquad (4)$$

$$\mathscr{X} = \begin{pmatrix} R_{11} \\ R_{21} \\ R_{31} \\ R_{22} \\ R_{23} \\ T_1 \\ T_2 \\ T_3 \end{pmatrix}, \mathscr{B} = \begin{pmatrix} 0 \\ -b_1 \\ 0 \\ -b_2 \\ 0 \\ 0 \end{pmatrix} \qquad (5)$$

The matrix $\mathscr{A}$ consists of known variables and is of rank 6. As there are 8 variables in the linear system we can obtain a solution in a subspace spanned by two vectors: $\mathscr{X} = \mathbf{u} + l_1\mathbf{v} + l_2\mathbf{w}$, where $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are known vectors of size 8×1. Next, we use orthogonality constraints from the rotation matrix to estimate the unknown variables l_1 and l_2. We can write two orthogonality constraints involving the rotation variables $R_{11}, R_{21}, R_{31}, R_{22}$, and R_{23}.

$$\begin{aligned}
R_{11}^2 + R_{21}^2 + R_{31}^2 &= 1 \\
R_{21}^2 + R_{22}^2 + R_{23}^2 &= 1
\end{aligned}$$

On substituting these rotation variables as functions of l_1 and l_2 and solving the above quadratic system of equations we obtain four solutions for (l_1, l_2) - thus, four solutions for $(R_{11}, R_{21}, R_{31}, R_{22}, R_{23})$. Using simple orthogonality constraints we can see that these five elements in the rotational matrix uniquely determine the other elements. Thus the 2 point and 1 line case gives a total of *four solutions* for the pose $(\mathsf{R}, \mathbf{T})$.



(a) (b)

(c) (d)

Fig. 4: *The choice of intermediate coordinate systems $\mathscr{C}_1$ and $\mathscr{W}_1$ for computing the pose using 1 point and 2 lines.*

B. 1 Point and 2 Lines

g) The choice of camera reference frame $\mathscr{C}_1$: In figure 4(a) and (b), we show the camera projection rays (associated with 2D points and lines) and 3D features (points and lines) in $\mathscr{C}_0$ and $\mathscr{W}_0$ respectively. In $\mathscr{C}_0$, let the center of the camera be C^0, the projection ray corresponding to the 2D point be given by direction vector $\vec{d}_1$, the projection rays corresponding to the two 2D lines be given by the pairs of direction vectors $(\vec{d}_2,\vec{d}_3)$ and $(\vec{d}_4,\vec{d}_5)$.

In $\mathscr{C}_1$, let the ray corresponding to the 2D point be given by CD_1, the rays linked with the two lines be given by pairs (CD_2,CD_3) and (CD_3,CD_4) respectively. We choose $\mathscr{C}_1$ satisfying the following conditions:

- The center of the camera is at $(0,0,-1)$.
- The projection ray CD_3 from the line of intersection of the two interpretation planes lie on the $\mathscr{Z}$ axis such that $D_3=(0,0,0)$.
- The ray CD_2 lies on the $\mathscr{X}\mathscr{Z}$ plane where D_2 is on $\mathscr{X}$ axis.

Similar to the previous case, we prove that such a transformation is possible by construction. The unit normal vectors for the interpretation planes $(C^0,\vec{d}_2,\vec{d}_3)$ and $(C^0,\vec{d}_4,\vec{d}_5)$ are given by $\vec{n}_1=\vec{d}_2\times\vec{d}_3$ and $\vec{n}_2=\vec{d}_4\times\vec{d}_5$. The direction vector of the line of intersection of the two planes can be computed as $\vec{d}_{12}=\vec{n}_1\times\vec{n}_2$. The direction vectors $\vec{d}_2$, $\vec{d}_{12}$ and $\vec{d}_4$ in $\mathscr{C}_0$ correspond to the projection rays CD_2, CD_3 and CD_4 respectively. Using simple algebraic transformations we show the points D_2 and D_3 before and after transformation to the intermediate reference frames:

$$
\begin{aligned}
D_2^0 &= C^0+\frac{\vec{d}_2}{\vec{d}_2.\vec{d}_{12}},\\
D_3^0 &= C^0+\vec{d}_{12},\\
D_2 &= \begin{pmatrix} \tan(\cos^{-1}(\vec{d}_2.\vec{d}_{12})) \\ 0 \\ 0 \end{pmatrix},\\
D_3 &= \mathbf{0}_{3\times1}
\end{aligned}
$$

The transformation between $\mathscr{C}_0$ and $\mathscr{C}_1$ is given by the one that maps the triplet (C^0,D_2^0,D_3^0) to (C,D_2,D_3).

h) The choice of world reference frame $\mathscr{W}_1$: The world reference frame $\mathscr{W}_1$ is chosen such the single 3D point P_1 lies at the origin $(0,0,0)$. The transformation between $\mathscr{W}_0$ and $\mathscr{W}_1$ is a simple translation that translates P_1^0 to P_1.

We use the following notation for the points in $\mathscr{C}_1$ and $\mathscr{W}_1$:

$$
D_{i=\{1,4\}}=\begin{pmatrix} a_i \\ b_i \\ 0 \end{pmatrix},\ D_3=\mathbf{0}_{3\times1},\ L_{i=\{1,2,3,4\}}=\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} \qquad (6)
$$

i) Pose estimation between $\mathscr{C}_1$ and $\mathscr{W}_1$: Now we show the pose estimation using one point and two line correspondences. We stack the two collinearity equations from the triplet (C,D_1,P_1) and four coplanarity equations from the quadruplets (C,D_2,D_3,L_1), (C,D_2,D_3,L_2), (C,D_3,D_4,L_3) and (C,D_3,D_4,L_4). We can build the following linear system: $\mathscr{A}\mathscr{X}=\mathscr{B}$, where $\mathscr{A}$, $\mathscr{X}$ and $\mathscr{B}$ are given below:

$$
\mathscr{A}=\begin{pmatrix}
0 & 0 & 0 & 0 & -b_4X_3 & -b_4X_4 \\
0 & 0 & 0 & 0 & -b_4Y_3 & -b_4Y_4 \\
0 & 0 & 0 & 0 & -b_4Z_3 & -b_4Z_4 \\
0 & 0 & X_1 & X_2 & a_4X_3 & a_4X_4 \\
0 & 0 & Y_1 & Y_2 & a_4Y_3 & a_4Y_4 \\
0 & 0 & Z_1 & Z_2 & a_4Z_3 & a_4Z_4 \\
-b_1 & 0 & 0 & 0 & -b_4 & -b_4 \\
a_1 & -1 & 1 & 1 & a_4 & a_4 \\
0 & b_1 & 0 & 0 & 0 & 0
\end{pmatrix}^T, \qquad (7)
$$

$$
\mathscr{X}=\begin{pmatrix} R_{11} \\ R_{12} \\ R_{13} \\ R_{21} \\ R_{22} \\ R_{23} \\ T_1 \\ T_2 \\ T_3 \end{pmatrix},\ \mathscr{B}=\begin{pmatrix} 0 \\ -b_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad (8)
$$

In the linear system $\mathscr{A}\mathscr{X}=\mathscr{B}$, the first and second rows are obtained using the collinearity constraint shown in equation (1) for the triplet (C,D_1,P_1). The third, fourth, fifth and sixth rows are obtained using the coplanarity constraint shown in equation (2) for the quadruplets (C,D_2,D_3,L_1), (C,D_2,D_3,L_2), (C,D_3,D_4,L_3) and (C,D_3,D_4,L_4) respectively. The matrix $\mathscr{M}$ consists of known variables and is of rank 6. As there are 9 variables in the linear system we can obtain a solution in a subspace spanned by three vectors: $\mathscr{X} = \mathbf{u}+l_1\mathbf{v}+l_2\mathbf{w}+l_3\mathbf{y}$, where $\mathbf{u},\mathbf{v},\mathbf{w}$ and $\mathbf{y}$ are known vectors of size 9×1 and l_1, l_2 and l_3 are unknown variables. We write three orthogonality constraints involving the rotation variables $R_{11},R_{12},R_{13},R_{21},R_{22}$ and R_{23} (individual elements in $\mathscr{X}$ expressed as functions of l_1, l_2 and l_3):

$$\begin{aligned}R_{11}^2+R_{12}^2+R_{13}^2 &= 1\\ R_{21}^2+R_{22}^2+R_{23}^2 &= 1\\ R_{11}R_{21}+R_{12}R_{22}+R_{13}R_{23} &= 0\end{aligned}$$

On solving the polynomial equation we obtain up to 8 different solutions for l_1. This leads to 8 solutions for both l_2 and l_3. Consequently, this produced eight solutions for the pose $(\mathsf{R},\mathbf{T})$. Note that pose estimation using three lines also gives 8 different solutions.

j) Degenerate cases and other scenarios: Among the 3D features, if a 3D point lies on a 3D line then the configuration is degenerate. It is possible to solve the three points and three lines using the same idea of coordinate transformation and the use of collinearity and coplanarity constraints.

IV. Experiments

k) Simulations: We designed a few synthetic experiments to quantify the performance of the various minimal algorithms for different noise levels. We generated projections of 10 points and 10 lines in the cube $[-1,1]^3$ for varying camera poses. We added Gaussian noise of zero mean and varying standard deviations for the different points in the image. In order to propagate the noise for the line parameters we used the technique proposed in [34]. We used 2000 trials to study the behavior of different algorithms - four minimal algorithms, two non-minimal ones and a *hybrid approach*. The hybrid approach refers to an algorithm that uses all four minimal algorithms developed by our framework. We randomly pick three features from all the point and line correspondences. Depending on the number of points and lines, we chose the corresponding algorithm. We used the sum of errors from both points and lines to select the best one from all the iterations. For points, reprojection error was used. In the case of lines, we used the same error metric as in [32].

We studied the rotation and translation error in the simulations, see figure 5. As expected, minimal solutions gave lower error compared to non-minimal ones [17], [1]. In the case of translation error, the method of [1] was still close to the minimal solutions. As the standard deviation of the noise increases, the mixed scenarios started giving lower error compared to non-mixed ones. Although our experiments suggested that minimal solutions give lower error compared to non-minimal ones, it is difficult to decide the best minimal algorithm. Our experiments suggested that 3 lines are better than 3 points in general. However in real scenarios, depending on the distribution and availability of points and lines, any one of the four minimal algorithms can outperform the rest.

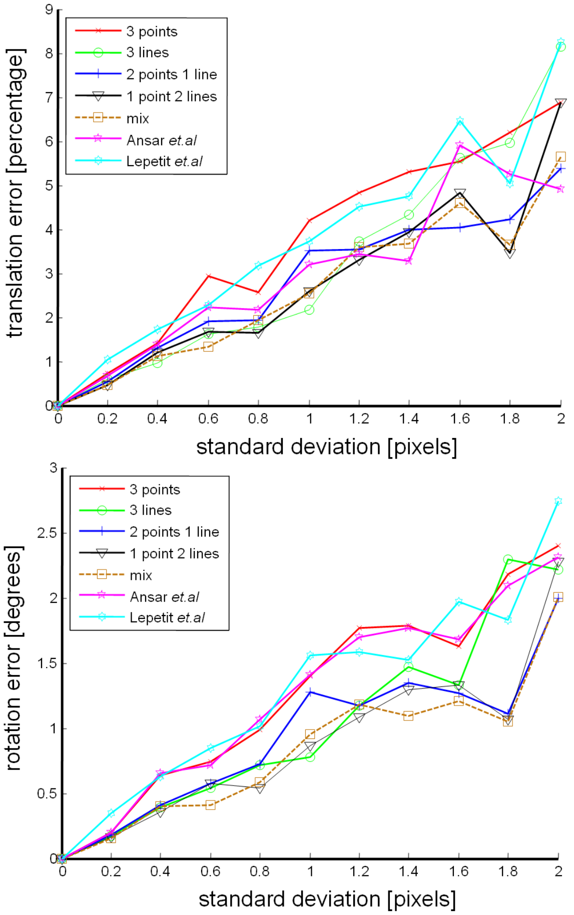


Fig. 5: *Noise simulations to study the translational (a) and rotational (b) error for various algorithms proposed in this paper and two other non-minimal algorithms.*

l) Geo-localization using coarse 3D models: We used coarse 3D models of Boston purchased from commercial websites[2]. These models are plane-based and does not have fine architectural details. Now we briefly explain our method to register a sequence of images to the 3D model, see also figure 6. We register the first image in the sequence with the 3D model by manually giving the 2D-3D correspondences. Then we obtain point and line correspondences between the first and the second images. By back-projecting the features (points and lines) from the first image on to the 3D model we obtain their 3D coordinates. Using this we can compute the 3D-2D correspondence between the 3D model and the second image. Next we use the hybrid approach to compute the pose of the second image. We continue this process to register a sequence of images to a coarse 3D model.

[2] http://www.3dcadbrowser.com/

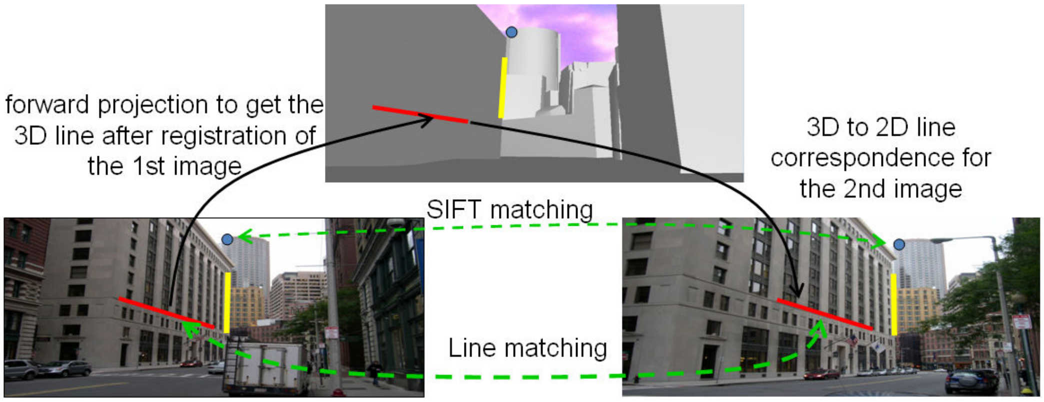

forward projection to get the 3D line after registration of the 1st image
3D to 2D line correspondence for the 2nd image
SIFT matching
Line matching

Fig. 6: *Point and line correspondences are computed between the first and the second image using SIFT descriptors. Knowing the registration of the first image, we obtain the 3D coordinates of these correspondences by back-projection to the 3D model. After these two steps the 2D-3D point and line correspondences are known for the second image and the new pose can be computed. This process is iteratively repeated to find the geolocalization of all the images in the data set.*

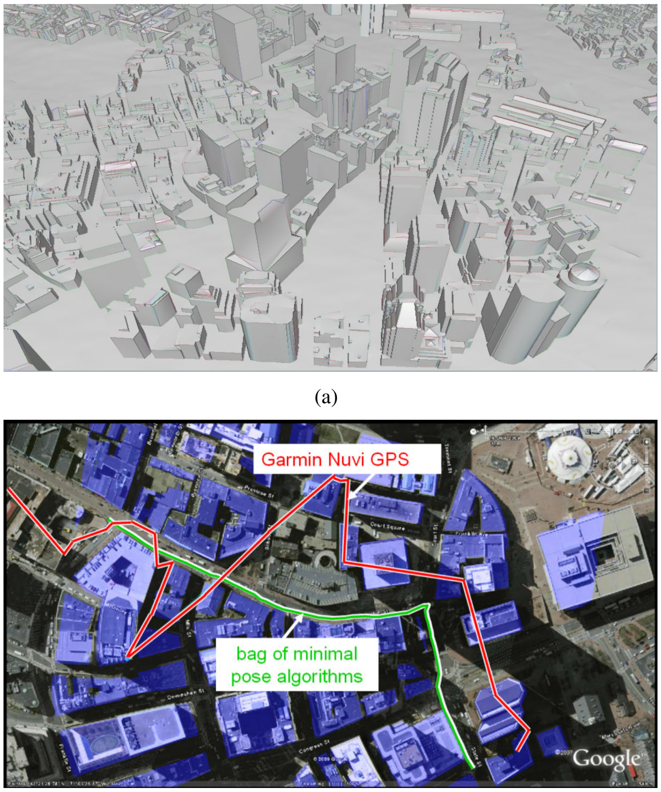
Note that the 3D lines need not always come from depth discontinuities in the scene. They can also be taken from the middle of a planar wall as shown in Figure 6.

About 177 images were tested in Boston's financial district and the results were promising, see figures 7 and 8. There were occasional slight mismatches for some lines because of the inaccuracies in the 3D model. However, the geo-localization is much better than the results of Garmin Nüvi 255W GPS estimates for the same region with tall buildings. The proposed algorithm is extremely suitable for really challenging scenarios with pedestrians, cars and missing buildings. Our method will be very useful for such scenarios and probably be the most robust one. In the Supplementary Materials we show a video of a geo-localization experiment in the Boston's Financial district.

V. Conclusion

In several real world applications finding three non-degenerate point or line correspondences is not always possible. Our work improves this situation by giving a choice of mixing these features and thereby enabling a solution in cases which were not possible before. Three point pose estimation has been used for outdoor SLAM algorithms. For indoor scenarios, 3-line pose estimation approaches are more robust due to the lack of discriminative feature points. We believe that our solutions can lead to SLAM algorithms that can work in both indoor and outdoor scenarios.

Acknowledgments: We would like to specially thank Jay Thornton for the valuable feedback and useful discussions throughout the project. We would also like to thank Yuichi Taguchi, Matthew Brand, Keisuke Kojima, Joseph Katz, Haruhisa Okuda, Hiroshi Kage, Kazuhiko Sumi, Daniel Thalmann, and the anonymous reviewers for their valuable feedback and support.


(a)

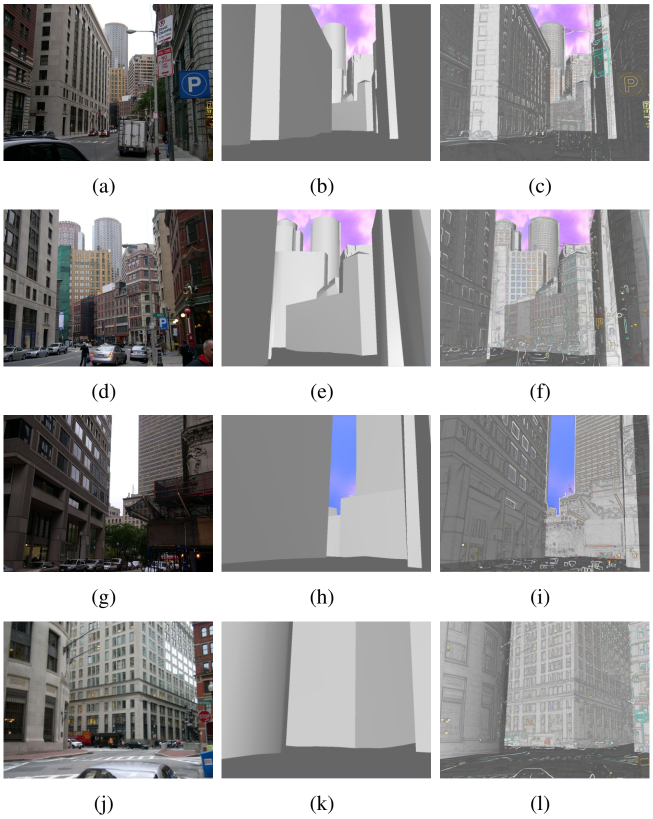
(b)

Fig. 7: *(a) The 3D model of Boston used for the geo-localization experiment. (b) Geo-localization comparison between our minimal approach and GPS Garmin Nüvi 255W*

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *PAMI*, 2003.

[2] T. Bonde and H. Nagel. Deriving a 3-d description of a moving rigid object from monocular tv-frame sequence. In *J.K Aggarwal and N.I. Badler, editor, Proc. Workshop on Computer Analysis of Time Varying Imagery*, 1979.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 8: *The first column shows the real images. The second column shows the rendering of the 3D model after geo-localization. Finally the third column shows the reprojection of the edges from real image on the 3D model after geo-localization.*

[3] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy image sequences. *PAMI*, 1986.

[4] M. Dhome, M. Richetin, J.-T. Lapresté, and G. Rives. Determination of the attitude of 3-D objects from a single perspective view. *PAMI*, 11(12):1265–1278, 1989.

[5] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 2006.

[6] C. S. (editor). *Manual of Photogrammetry*. Fourth Edition, ASPRS, 1980.

[7] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[8] X. Gao, X. Hou, J. Tang, and H. Cheng. Complete solution classification for the perspective-three-point problem. *PAMI*, 2003.

[9] C. Geyer and H. Stewenius. A nine-point algorithm for estimating para-catadioptric fundamental matrices. In *CVPR*, 2007.

[10] J. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in der Geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1:238248, 1841.

[11] R. Haralick, C. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision (IJCV)*, 13(3):331–356, 1994.

[12] J. Hays and A. Efros. Im2gps: estimating geographic images from single images. In *CVPR*, 2008.

[13] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society A*, 4(4):629–642, April 1987.

[14] N. Jacobs, S. Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *ICCV*, 2007.

[15] O. Koch and S. Teller. Wide-area egomotion estimation from known 3d structure. In *CVPR*, 2007.

[16] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. In *CVPR*, 2007.

[17] V. Lepetit, F. Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *IJCV*, 2008.

[18] H. Li and R. Hartley. A non-iterative method for correcting lens distortion from nine-point correspondenses. In *OMNIVIS*, 2005.

[19] D. Nistér. An efficient solution to the five-point relative pose problem. In *CVPR*, 2003.

[20] D. Nister, R. Hartley, and H. Stewanius. Using galois theory to prove structure from motion algorithms are optimal. In *CVPR*, 2007.

[21] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 2006.

[22] C. Olsson, F. Kahl, and M. Oskarsson. The registration problem revisited: Optimal solutions from points, lines and planes. In *CVPR*, volume 1, pages 1206–1213, June 2006.

[23] S. Ramalingam*, S. Bouaziz*, P. Sturm, and M. Brand. Skyline2gps: Localization in urban canyons using omni-skylines. In *IROS*, 2010.

[24] S. Ramalingam, P. Sturm, and S. Lodha. Towards complete generic camera calibration. In *CVPR*, 2005.

[25] S. Ramalingam, Y. Taguchi, T. Marks, and O. Tuzel. P2pi: A minimal solution for the registration of 3d points to 3d planes. In *ECCV*, 2010.

[26] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.

[27] E. Royer, M. Lhuillier, and M. Dhome. Monocular vision for mobile robot localization. *IJCV*, 2007.

[28] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3d. In *SIGGRAPH*, 2006.

[29] H. Stewenius, D. Nister, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *CVPR*, 2005.

[30] H. Stewenius, D. Nister, M. Oskarsson, and K. Astrom. Solutions to minimal generalized relative pose problems. In *OMNIVIS*, 2005.

[31] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 2011.

[32] C. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. *PAMI*, 1995.

[33] T. Yeh, K. Tollmar, and T. Darrell. Searching the web with mobile images for location recognition. In *CVPR*, 2004.

[34] S. Yi, R. Haralick, and L. Shapiro. Error propagation in machine vision. *Machine vision and applications*, 1994.

[35] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.