plane-line-SLAM

Sun Qinxuan

August 10, 2020

1 Plane and Line Extraction

The planes are extracted using STING-based plane extraction method proposed in [1] and are denoted by $\pi = [\boldsymbol{n}^T, d]^T$, where $\boldsymbol{n}$ is the unit normal and d is the vertical distance from the origin to the plane. The mean and covariance of the RGB values corresponding to the points on the plane π are computed and denoted by $\boldsymbol{r}_\pi$ and $\boldsymbol{S}_\pi$, respectively.

The lines are detected in the image by LSD algorithm [2] and projected to the camera coordinate system. $\mathcal{L} = [\boldsymbol{u}^T, \boldsymbol{v}^T]^T$ represent the parameters of the line feature, where $\boldsymbol{u} \in \mathbb{R}^3$ is a vector with its direction orthogonal to the plane defined by the join of the line and the origin, and its norm equal to the vertical distance from the origin to the line, and $\boldsymbol{v} \in \mathbb{S}^2$ is the unit direction vector of the line.

The coordinate system in which the features are described is denoted by the subscript. Specifically, the current and reference frames are denoted by the subscripts c and r, respectively.

2 Plane-Line Association Graph (PLAG)

The plane association graph (PAG) is proposed in [1] to set up correspondences between planes extracted from two successive scans. The PAG is a graph built for each scan, wherein nodes represent the extracted planes and edges represent the geometric relationships between the planes. In this section, we extend the PAG to associate plane and line features simultaneously, which yields the plane-line association graph (PLAG).

2.1 Construction of PLAG

For the set of planes $\{\pi_i\}_{i=1,\cdots,N_\pi}$ and lines $\{\mathcal{L}_j\}_{j=1,\cdots,N_\mathcal{L}}$ extracted from one frame, the PLAG $G = (V, E)$ is built with V representing the set of

vertices and E the set the edges. There are two different kinds of vertices in V, i.e., the plane vertex $v_{\pi,i} = \pi_i, i = 1, \cdots, N_\pi$ and the line vertex $v_{\mathcal{L},j} = \boldsymbol{\mathcal{L}}_j, j = 1, \cdots, N_\mathcal{L}$. Accordingly, E also includes two types of edges: plane-plane edge $e_{\pi,ik}$ and line-plane edge $e_{\mathcal{L},jk}$. The plane-plane edge $e_{\pi,ik}$ connects two plane vertices and the line-plane edge $e_{\mathcal{L},jk}$ connects a line vertex and a plane vertex.

$$e_{\pi,ik} = (\omega_{\pi,ik}, \alpha_{\pi,ik}, d_{\pi,ik}), i, k \in \{1, \cdots, N_\pi\}, i \neq k, \tag{1}$$

$$e_{\mathcal{L},jk} = (\omega_{\mathcal{L},jk}, \alpha_{\mathcal{L},jk}, d_{\mathcal{L},jk}), j \in \{1, \cdots, N_\mathcal{L}\}, k \in \{1, \cdots, N_\pi\}. \tag{2}$$

The two types of edges $e_{\pi,ik}$ and $e_{\mathcal{L},jk}$ encode the geometric relationships between two planes and between a line and a plane , respectively. The geometric relationships can be classified into two categories: parallel and non-parallel, which are indicated by the variables $\omega_{\pi,ik}$ and $\omega_{\mathcal{L},jk}$ for $e_{\pi,ik}$ and $e_{\mathcal{L},jk}$, respectively.

$$\omega_{\pi,ik} = \begin{cases} parallel, & \text{if } \alpha_{\pi,ik} < \delta_{rad}; \\ non-parallel, & \text{otherwise.} \end{cases} \tag{3}$$

$$\omega_{\mathcal{L},jk} = \begin{cases} parallel, & \text{if } \alpha_{\mathcal{L},ij} < \delta_{rad}; \\ non-parallel, & \text{otherwise.} \end{cases} \tag{4}$$

$\alpha_{\pi,ik}$ represents the angle between $\boldsymbol{n}_i$ and $\boldsymbol{n}_k$ and $\alpha_{\mathcal{L},jk}$ the angle between $\boldsymbol{v}_j$ and $\boldsymbol{n}_k$.

$$\alpha_{\pi,ik} = \arccos\left(\boldsymbol{n}_i^T \boldsymbol{n}_k\right), \tag{5}$$

$$\alpha_{\mathcal{L},jk} = \arccos\left(\boldsymbol{v}_j^T \boldsymbol{n}_k\right). \tag{6}$$

$d_{\pi,ik}$ and $d_{\mathcal{L},jk}$ are defined, respectively, by

$$d_{\pi,ik} = \begin{cases} |d_i - d_k|, & \text{if } \omega_{\pi,ik} = parallel; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$$d_{\mathcal{L},jk} = \begin{cases} |\boldsymbol{n}_k^T [\boldsymbol{v}_j]_\times \boldsymbol{u}_j + d_k|, & \text{if } \omega_{\mathcal{L},jk} = parallel; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

According to (7) and (8), when two planes (a line and a plane) are parallel to each other, $d_{\pi,ik}$ ($d_{\mathcal{L},jk}$) represent the vertical distance between the two planes (the line and plane).

2.2 Feature Matching using PLAG

For the two feature sets $\{\pi_{ci}, \mathcal{L}_{cj}\}_{i=1,\cdots,N_{\pi c}, j=1,\cdots,N_{\mathcal{L}c}}$ and $\{\pi_{rm}, \mathcal{L}_{rn}\}_{m=1,\cdots,N_{\pi r}, n=1,\cdots,N_{\mathcal{L}r}}$ extracted from the current and reference scans, respectively, two PLAGs $G_c = (V_c, E_c)$ and $G_r = (V_r, E_r)$ are constructed, respectively, using the method proposed in Section 2.1. Then, the features from two scans are associated by calculating the similarities of vertices from G_c and G_r.

First, the relationships between two plane-plane edges and between two plane-line edges are defined, respectively, as

$$\begin{cases} e_{\pi c,ik} = e_{\pi r,ml} & \text{if } \omega_{\pi c,ik} = \omega_{\pi r,ml} \text{ and } |\alpha_{\pi c,ik} - \alpha_{\pi r,ml}| < \delta_{rad} \\ & \text{and } |d_{\pi c,ik} - d_{\pi r,ml}| < \delta_{dist}, \\ e_{\pi c,ik} \neq e_{\pi r,ml} & \text{otherwise.} \end{cases} \tag{9}$$

$$\begin{cases} e_{\mathcal{L}c,jk} = e_{\mathcal{L}r,nl} & \text{if } \omega_{\mathcal{L}c,jk} = \omega_{\mathcal{L}r,nl} \text{ and } |\alpha_{\mathcal{L}c,jk} - \alpha_{\mathcal{L}r,nl}| < \delta_{rad} \\ & \text{and } |d_{\mathcal{L}c,jk} - d_{\mathcal{L}r,nl}| < \delta_{dist}, \\ e_{\mathcal{L}c,jk} \neq e_{\mathcal{L}r,nl} & \text{otherwise.} \end{cases} \tag{10}$$

Then, the similarity for two vertices of the same type is defined for plane vertices and line vertices, respectively. Specifically, for two plane vertices $v_{\pi c,i} \in V_c$ and $v_{\pi r,m} \in V_r$, the similarity $s_\pi(v_{\pi c,i}, v_{\pi r,m})$ is defined by

$$s_\pi(v_{\pi c,i}, v_{\pi r,m}) = s_{\pi,\mathrm{col}}(v_{\pi c,i}, v_{\pi r,m}) + s_{\pi,\mathrm{geo}}(v_{\pi c,i}, v_{\pi r,m}), \tag{11}$$

where $s_{\pi,\mathrm{col}}(v_{\pi c,i}, v_{\pi r,m})$ and $s_{\pi,\mathrm{geo}}(v_{\pi c,i}, v_{\pi r,m})$ represent the color and geometric similarities, respectively. The color similarity $s_{\pi,\mathrm{col}}(v_{\pi c,i}, v_{\pi r,m})$ is defined as the Bhattacharyya distance between the color distributions of two planes.

$$s_{\pi,\mathrm{col}}(v_{\pi c,i}, v_{\pi r,m}) = \frac{1}{8}\left(\boldsymbol{r}_{\pi c,i} - \boldsymbol{r}_{\pi r,m}\right)^T \boldsymbol{S}_\pi^{-1}\left(\boldsymbol{r}_{\pi c,i} - \boldsymbol{r}_{\pi r,m}\right) + \frac{1}{2}\ln\left(\frac{|\boldsymbol{S}_\pi|}{\sqrt{|\boldsymbol{S}_{\pi c,i}| \cdot |\boldsymbol{S}_{\pi r,m}|}}\right), \tag{12}$$

where $\boldsymbol{S}_\pi = \frac{1}{2}\boldsymbol{S}_{\pi c,i} + \boldsymbol{S}_{\pi r,m}$. The geometric similarity $s_{\pi,\mathrm{geo}}(v_{\pi c,i}, v_{\pi r,m})$ is calculated by the sum of color similarities between the vertices connected to $v_{\pi c,i}$ and $v_{\pi r,m}$, respectively, by similar edges defined in (9).

$$s_{\pi,\mathrm{geo}}(v_{\pi c,i}, v_{\pi r,m}) = \frac{1}{|I_{im}|}\sum_{(k,l)\in I_{im}} s_{\pi,\mathrm{col}}(v_{\pi c,k}, v_{\pi r,l}). \tag{13}$$

I_{im} is a set of index pairs and is defined by

$$I_{im} = \{(k,l) | e_{\pi c,ik} = e_{\pi r,ml}, k \in 1,\cdots,N_{\pi c}, k \neq i, l \in 1,\cdots,N_{\pi r}, l \neq m.\}. \tag{14}$$

And $|I_{im}|$ represents the number of index pairs in I_{im}.

For two line vertices $v_{\mathcal{L}c,j} \in V_c$ and $v_{\mathcal{L}r,n} \in V_r$, the similarity $s_{\mathcal{L}}(v_{\mathcal{L}c,j}, v_{\mathcal{L}r,n})$ is defined by

$$s_{\mathcal{L}}(v_{\mathcal{L}c,j}, v_{\mathcal{L}r,n}) = s_{\mathcal{L},\text{geo}}(v_{\mathcal{L}c,j}, v_{\mathcal{L}r,n}), \tag{15}$$

where the geometric similarity $s_{\mathcal{L},\text{geo}}(v_{\mathcal{L}c,j}, v_{\mathcal{L}r,n})$ is calculated by the sum of color similarities between the plane vertices connected to $v_{\mathcal{L}c,j}$ and $v_{\mathcal{L}r,n}$, respectively, by similar edges defined by (10).

$$s_{\mathcal{L},\text{geo}}(v_{\mathcal{L}c,j}, v_{\mathcal{L}r,n}) = \frac{1}{|I_{jn}|} \sum_{(k,l)\in I_{jn}} s_{\pi,\text{col}}(v_{\pi c,k}, v_{\pi r,l}). \tag{16}$$

I_{jn} is a set of index pairs and is defined by

$$I_{jn} = \{(k,l) | e_{\mathcal{L}c,jk} = e_{\mathcal{L}r,nl}, k \in 1, \cdots, N_{\pi c}, l \in 1, \cdots, N_{\pi r}.\}. \tag{17}$$

And $|I_{jn}|$ represents the number of index pairs in I_{jn}.

Each pair of planes and lines from two frames is examined by the similarities (11) and (15), respectively, to find correspondence between them.

3 plane-line-VO

In this section, the plane-line-VO is proposed to estimate the 6-DoF RGB-D camera pose $\mathbf{R}$ and $\mathbf{t}$. For RGB-D sensors, plane features are more stably extracted than line features because the lines are normally detected along the edges of objects, where both the RGB and depth measurements are more noisy than those on flat surfaces [3] [4]. Therefore, we first compute the camera motion using parameters of the associated plane features, as in [1]. However, the quantity of planes extracted from the scan is relatively small and it frequently occurs that estimation of the camera pose cannot be fully constrained. In [1], the degenerate cases are detected according to the spatial configuration of the plane features and a STING-based scan matching algorithm is proposed to fully constrain the motion. Nevertheless, the scan matching is an iterative process and might be trapped in a local minimum if the camera moves fast. In this section, when plane features are insufficient to constrain the motion estimation, we use the line features together with the planes to obtain an accurate and robust result.

The associated plane and line pairs from two frames are denoted by $\{\pi_{ci}, \pi_{ri}\}_{i=1,\cdots,N_\pi}$ and $\{\mathcal{L}_{cj}, \mathcal{L}_{rj}\}_{j=1,\cdots,N_\mathcal{L}}$, respectively. As presented in [1], the rotation $\boldsymbol{R} \in \mathbb{SE}(3)$ and translation $\boldsymbol{t} \in \mathbb{R}^3$ of the camera that best align the matched

planes are obtained by minimizing (18) and (19), respectively.

$$J_{\pi R}(\boldsymbol{R}) = \sum_{i=1}^{N_\pi} \|\boldsymbol{n}_{ci} - \boldsymbol{R}\boldsymbol{n}_{ri}\|^2, \tag{18}$$

$$J_{\pi t}(\boldsymbol{t}) = \sum_{i=1}^{N_\pi} \left(d_{ci} - (d_{ri} - \boldsymbol{n}_{ci}^T\boldsymbol{t})\right)^2. \tag{19}$$

To distinguish the degenerate cases when using plane features, a matrix $\boldsymbol{H} = \sum_{i=1}^{N_\pi} \boldsymbol{n}_{ri}\boldsymbol{n}_{ci}^T$ is defined and the singular value decomposition (SVD) is performed for $\boldsymbol{H}$.

$$\boldsymbol{H} = \boldsymbol{Q}_r\boldsymbol{\Lambda}\boldsymbol{Q}_c^T = \lambda_1\boldsymbol{q}_{r1}\boldsymbol{q}_{c1}^T + \lambda_2\boldsymbol{q}_{r2}\boldsymbol{q}_{c2}^T + \lambda_3\boldsymbol{q}_{r3}\boldsymbol{q}_{c3}^T, \tag{20}$$

where $\boldsymbol{Q}_r = [\boldsymbol{q}_{r1}, \boldsymbol{q}_{r2}, \boldsymbol{q}_{r3}]$ and $\boldsymbol{Q}_c = [\boldsymbol{q}_{c1}, \boldsymbol{q}_{c2}, \boldsymbol{q}_{c3}]$ are orthonormal matrices, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \lambda_3\}, \lambda_1 \geq \lambda_2 \geq \lambda_3$. Three kinds of cases, i.e., one constrained case and two degenerate cases, can be distinguished according to the singularity values of $\boldsymbol{H}$.

3.1 6-DoF Constraint Case

If $\boldsymbol{H}$ is nonsingular, i.e., $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$, the rotation that minimizes (18) can be calculated by

$$\hat{\boldsymbol{R}} = \boldsymbol{Q}_c\boldsymbol{Q}_r^T. \tag{21}$$

And the translation that minimize (19) can be obtained through the least-squares method.

$$\hat{\boldsymbol{t}} = (\boldsymbol{A}_\pi^T\boldsymbol{A}_\pi)^{-1}\boldsymbol{A}_\pi^T\boldsymbol{d}_\pi, \tag{22}$$

where

$$\boldsymbol{A}_\pi = \begin{bmatrix} \boldsymbol{n}_{c1}^T \\ \vdots \\ \boldsymbol{n}_{cN_\pi}^T \end{bmatrix}, \boldsymbol{d}_\pi = \begin{bmatrix} d_{r1} - d_{c1} \\ \vdots \\ d_{rN_\pi} - d_{cN_\pi} \end{bmatrix}. \tag{23}$$

3.2 5-DoF Constraint Case

If $\lambda_1 \geq \lambda_2 > \lambda_3 = 0$, then $\{\boldsymbol{n}_{ci}\}_{i=1,\cdots,N_\pi}$ ($\{\boldsymbol{n}_{ri}\}_{i=1,\cdots,N_\pi}$) are coplanar and vertical to the vector $\boldsymbol{q}_{c3}$ ($\boldsymbol{q}_{r3}$). In this case, $\hat{\boldsymbol{R}}$ is computed by

$$\hat{\boldsymbol{R}} = \begin{cases} \boldsymbol{Q}_c\boldsymbol{Q}_r^T & \text{if } \det(\boldsymbol{Q}_c\boldsymbol{Q}_r^T) = 1; \\ \boldsymbol{Q}_c'\boldsymbol{Q}_r^T & \text{if } \det(\boldsymbol{Q}_c\boldsymbol{Q}_r^T) = -1. \end{cases} \tag{24}$$

where $\boldsymbol{Q}_c' = [\boldsymbol{q}_{c1}, \boldsymbol{q}_{c2}, -\boldsymbol{q}_{c3}]$.

The translation vector $\hat{\boldsymbol{t}}$ cannot be directly solved by (22) because $\det(\boldsymbol{A}_\pi^T \boldsymbol{A}_\pi) = 0$. In this case, the null space of the matrix $\boldsymbol{A}_\pi$ can be represented by

$$\text{null}(\boldsymbol{A}_\pi) = \eta \boldsymbol{q}_{c3}, \eta \in \mathbb{R}. \tag{25}$$

As a result, the translation along $\boldsymbol{q}_{c3}$ cannot be constrained by the planes. In this case, to fully constrain the translation, the cost function for the translation can be defined as

$$J_t(\boldsymbol{t}) = J_{\pi t}(\boldsymbol{t}) + J_{\mathcal{L}t}(\boldsymbol{t}), \tag{26}$$

with

$$J_{\mathcal{L}t}(\boldsymbol{t}) = \sum_{j=1}^{N_\mathcal{L}} w_j \|\boldsymbol{u}_{cj} - (\hat{\boldsymbol{R}}\boldsymbol{u}_{rj} + [\boldsymbol{v}_{cj}]_\times \boldsymbol{t})\|^2. \tag{27}$$

The weight w_j is computed by

$$w_j = \|\boldsymbol{v}_{cj} \times \boldsymbol{q}_{c3}\|. \tag{28}$$

Note that, for $\boldsymbol{t} \in \text{null}(\boldsymbol{A}_\pi)$, the line vertical to $\boldsymbol{t}$ provide the largest constraint, which is assigned the largest weight according to (28). Then, the least-squares solution to the translation is

$$\hat{\boldsymbol{t}} = (\boldsymbol{A}^T \boldsymbol{W} \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{W} \boldsymbol{d}, \tag{29}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_\pi \\ \boldsymbol{A}_\mathcal{L} \end{bmatrix}, \boldsymbol{d}_\pi = \begin{bmatrix} \boldsymbol{d}_\pi \\ \boldsymbol{d}_\mathcal{L} \end{bmatrix}, \tag{30}$$

$$\boldsymbol{A}_\mathcal{L} = \begin{bmatrix} [\boldsymbol{v}_{c1}]_\times \\ \vdots \\ [\boldsymbol{v}_{cN_\mathcal{L}}]_\times \end{bmatrix}, \boldsymbol{d}_\mathcal{L} = \begin{bmatrix} \hat{\boldsymbol{R}}\boldsymbol{u}_{rj} - \boldsymbol{u}_{cj} \\ \vdots \\ \hat{\boldsymbol{R}}\boldsymbol{u}_{rN_\mathcal{L}} - \boldsymbol{u}_{cN_\mathcal{L}} \end{bmatrix}, \tag{31}$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{I}_{N_\pi} & & & \\ & w_1 \boldsymbol{I}_3 & & \\ & & \ddots & \\ & & & w_{N_\mathcal{L}} \boldsymbol{I}_3 \end{bmatrix}. \tag{32}$$

$\boldsymbol{I}_n$ is an n-dimensional identity matrix.

3.3 3-DoF Constraint Case

If $\lambda_1 > \lambda_2 = \lambda_3 = 0$, then $\{\boldsymbol{n}_{ci}\}_{i=1,\cdots,N_\pi}$ ($\{\boldsymbol{n}_{ri}\}_{i=1,\cdots,N_\pi}$) are colinear and are along the direction of $\boldsymbol{q}_{c1}$ ($\boldsymbol{q}_{r1}$).

In this case, the rotation about $\boldsymbol{q}_{c1}$ cannot be constrained only by planes, and there exist infinite rotation matrices that can minimize (18). To obtain a unique solution, the cost function for the rotation is defined as

$$J_R(\boldsymbol{R}) = J_{\pi R}(\boldsymbol{R}) + J_{\mathcal{L}R}(\boldsymbol{R}), \tag{33}$$

$$J_{\mathcal{L}R}(\boldsymbol{R}) = \sum_{j=1}^{N_\mathcal{L}} \|\boldsymbol{v}_{cj} - \boldsymbol{R}\boldsymbol{v}_{rj}\|^2. \tag{34}$$

Similarly, define a matrix $\boldsymbol{H}_\mathcal{L} = \sum_{i=1}^{N_\pi} \boldsymbol{n}_{ri}\boldsymbol{n}_{ci}^T + \sum_{j=1}^{N_\mathcal{L}} \boldsymbol{v}_{rj}\boldsymbol{v}_{cj}^T$ and compute the SVD as

$$\boldsymbol{H}_\mathcal{L} = \boldsymbol{Q}_{\mathcal{L}r}\boldsymbol{\Lambda}_\mathcal{L}\boldsymbol{Q}_{\mathcal{L}c}^T = \lambda_{\mathcal{L}1}\boldsymbol{q}_{\mathcal{L}r1}\boldsymbol{q}_{\mathcal{L}c1}^T + \lambda_{\mathcal{L}2}\boldsymbol{q}_{\mathcal{L}r2}\boldsymbol{q}_{\mathcal{L}c2}^T + \lambda_{\mathcal{L}3}\boldsymbol{q}_{\mathcal{L}r3}\boldsymbol{q}_{\mathcal{L}c3}^T, \tag{35}$$

where $\boldsymbol{Q}_{\mathcal{L}r} = [\boldsymbol{q}_{\mathcal{L}r1}, \boldsymbol{q}_{\mathcal{L}r2}, \boldsymbol{q}_{\mathcal{L}r3}]$ and $\boldsymbol{Q}_{\mathcal{L}c} = [\boldsymbol{q}_{\mathcal{L}c1}, \boldsymbol{q}_{\mathcal{L}c2}, \boldsymbol{q}_{\mathcal{L}c3}]$ are orthonormal matrices, and $\boldsymbol{\Lambda}_\mathcal{L} = \mathrm{diag}\{\lambda_{\mathcal{L}1}, \lambda_{\mathcal{L}2}, \lambda_{\mathcal{L}3}\}, \lambda_{\mathcal{L}1} \geq \lambda_{\mathcal{L}2} \geq \lambda_{\mathcal{L}3}$. Then, the rotation that minimizes (33) is calculated by

$$\hat{\boldsymbol{R}} = \boldsymbol{Q}_{\mathcal{L}c}\boldsymbol{Q}_{\mathcal{L}r}^T. \tag{36}$$

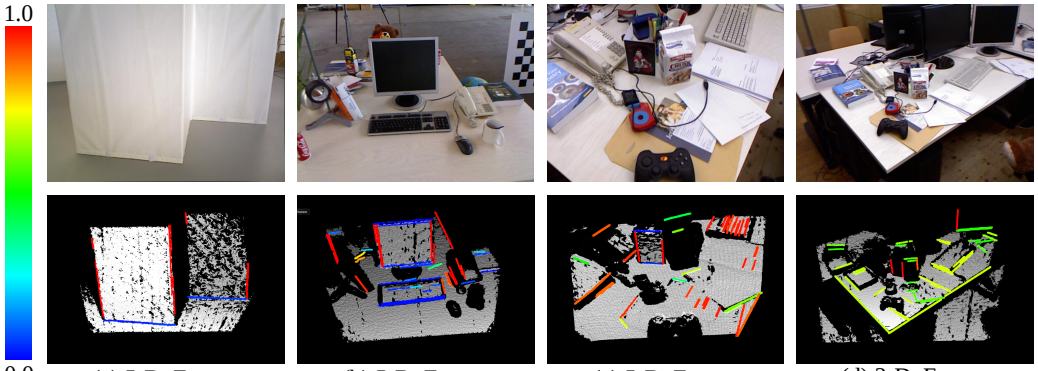
Similar to the 5-DoF case, the translation cannot be directly solved by (22), and in this case, the null space of the matrix $\boldsymbol{A}_\pi$ can be represented by

$$\mathrm{null}(\boldsymbol{A}_\pi) = \eta_2\boldsymbol{q}_{c2} + \eta_3\boldsymbol{q}_{c3}, \eta_2, \eta_3 \in \mathbb{R}. \tag{37}$$

The cost function for the translation is the same as (26), and the weight w_j is computed by

$$w_j = \frac{1}{2}\left(\|\boldsymbol{v}_{cj} \times \boldsymbol{q}_{c2}\| + \|\boldsymbol{v}_{cj} \times \boldsymbol{q}_{c3}\|\right). \tag{38}$$

And the optimal solution to the translation is computed as in (29).



(a) 5-DoF case (b) 5-DoF case (c) 5-DoF case (d) 3-DoF case

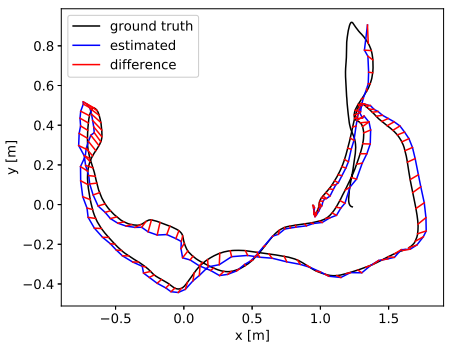
Figure 1: The weight assigned to each line feature are shown by a color map.

Table 1: Comparison of RPE RMSE.

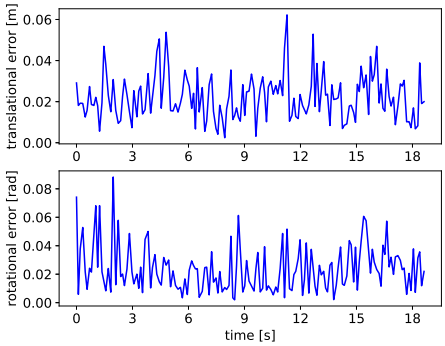
	fr1/desk	fr2/xyz	fr3/cabinet	fr3/str_ntex_far
plane-seg-VO	–	0.005m/0.36deg	0.034m/2.04deg	–
Prob-RGBD-VO	0.023m/1.70deg	–	0.039m/1.80deg	0.019m/0.70deg
Canny-VO	0.031m/1.92deg	**0.004m/0.31deg**	0.036m/1.63deg	0.027m/0.59deg
STING-VO	0.025m/1.90deg	**0.004m**/0.34deg	**0.011m/1.02deg**	0.014m/0.83deg
plane-line-VO	**0.022m/1.68deg**	**0.004m/0.31deg**	0.031m/1.30deg	**0.012m/0.50deg**

Table 2: Comparison of ATE RMSE.

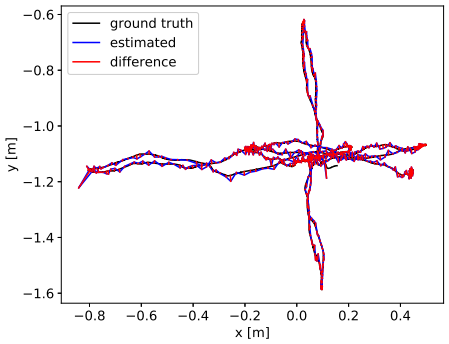
	fr1/desk	fr2/xyz	fr3/cabinet	fr3/str_ntex_far
Prob-RGBD-VO	**0.040m**	–	0.200m	0.054m
Canny-VO	0.044m	**0.008m**	0.057m	0.031m
STING-VO	0.041m	0.010m	0.070m	0.040m
plane-line-VO	**0.040m**	**0.008m**	**0.056m**	**0.027m**



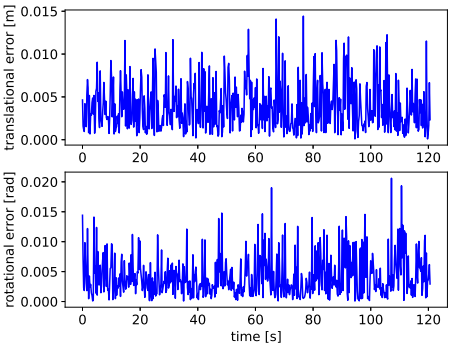
(a) ATE on fr1/desk



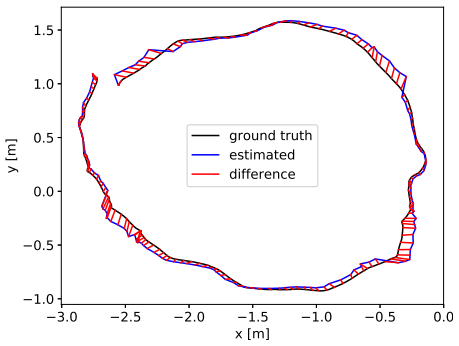
(b) RPE on fr1/desk



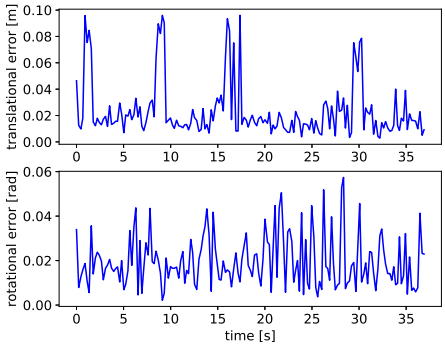
(c) ATE on fr2/xyz



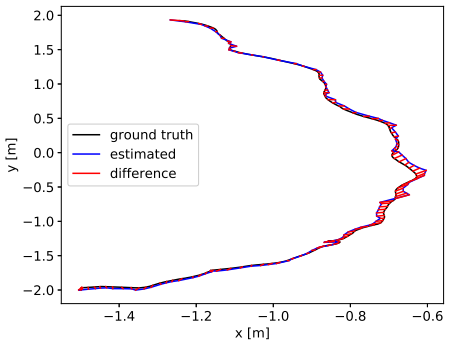
(d) RPE on fr2/xyz



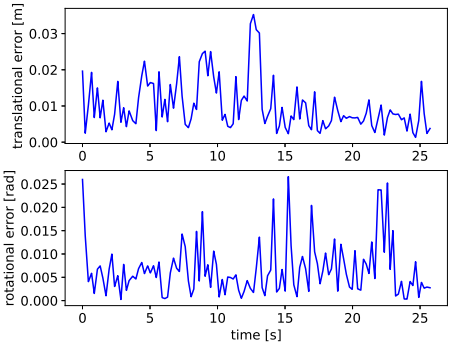
(e) ATE on fr3/cabinet



(f) RPE on fr3/cabinet



(g) ATE on fr3/str_ntex_far



(h) RPE on fr3/str_ntex_far

Figure 2: .

References

[1] Q. Sun, J. Yuan, X. Zhang, and F. Sun, "RGB-D SLAM in indoor environments with STING-based plane feature extraction," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1071–1082, 2018.

[2] R. Grompone von Gioi, J. Jakubowicz, J. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 722–732, April 2010.

[3] P. F. Proena and Y. Gao, "Probabilistic RGB-D odometry based on points, lines and planes under depth uncertainty," *Robotics and Autonomous Systems*, vol. 104, pp. 25 – 39, 2018.

[4] I. Dryanovski, R. G. Valenti, and Jizhong Xiao, "Fast visual odometry and mapping from RGB-D data," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2305–2310, 2013.