Plane-Line SLAM

Sun Qinxuan

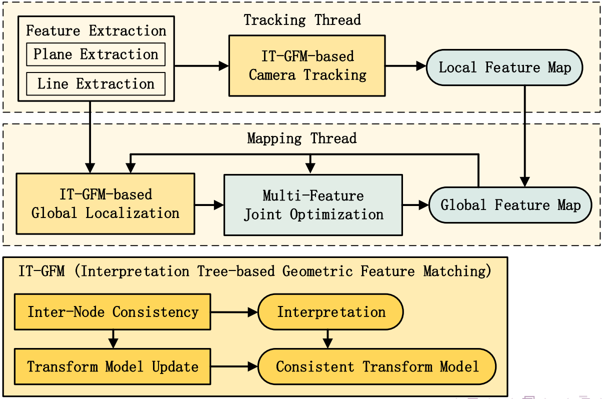
December 27, 2019

OUTLINE

1. System Overview
2. Interpretation Tree-based Geometric Feature Matching
   - Problem Formulation
   - Inter-Node Consistency
     - Rotation Consistency
     - Translation Consistency
   - Consistent Transform Model
3. Multi-Feature Joint Optimization
   - Parameterization for Planes and Lines
   - Computation of Information Matrices
4. Experiments

Introduction

- Plane-line-based SLAM System
  - Interpretation Tree-based Geometric Feature Matching (**IT-GFM**).
  - Multi-Feature Joint Optimization.



Introduction

- Contributions
  (1) An interpretation tree-based geometric feature matching (IT-GFM) is proposed combining plane and line features in a unified framework to associate features and estimate the pose of camera. IT-GFM generates closed-form solutions for the camera poses and does not involve any iterative optimization process. Therefore, it does not need any initial guess to guarantee the correct result and can be used in both camera tracking and the global localization tasks.
  (2) Inter-node consistency is proposed as a new criterion to generate the interpretations in the IT. It is designed based on the analysis of the constraints on the camera pose that are provided by two pairs of geometric features. The set of frame-to-frame transforms that aligns the two pairs of associated features are explicitly represented. Compared with the traditional pruning strategy in the IT-based method, the inter-node consistency guarantees that all the connected nodes in the IT can be aligned by some explicitly described rigid transform in 3D space. Thus, incorrect hypotheses are inherently avoided and the rigid transform information is further used in the update of the consistent transform model for the whole interpretation.

Introduction

- Contributions
  (3) A consistent transform model are designed for each interpretation to maintain a globally consistent transform for the interpretation. This transform model consists of all the nodes in an interpretation as well as a (or a set of) 3D rigid body transform that aligns all these nodes. It can be incrementally updated when a new node is added to the interpretation. With the increasing of the interpretation, more constraints are added to the frame-to-frame transform, which are explicitly expressed in the update of the transform model. Thus, a consistent transform is obtained when an interpretation is generated and is used as an initial value for the joint optimization.
  (4) Multi-feature joint optimization is implemented using the plane and line features. Both planes and lines are skillfully parameterized to maintain a globally consistent feature map. The information matrices for the two features are computed to well balance the two different components in the joint optimization process.

Interpretation Tree-based Geometric Feature Matching (IT-GFM)

- Notations
  - A geometric feature is represented by $\mathcal{F} \in \{\pi, \mathcal{L}\}$.
  - Features in the current frame: $\{\mathcal{F}_{ci}\}_{i=1,\cdots,N_c}$.
  - Features in the reference frame: $\{\mathcal{F}_{rj}\}_{j=1,\cdots,N_r}$.
  - The subscript c and r denote current and reference frame, respectively. And the subscript number i denote the index of the feature extracted from one scan.
  - A node in the IT: $\mathcal{N} = (\mathcal{F}_c, \mathcal{F}_r)$.
  - An n-interpretation in the IT: $\mathcal{P}_n = \{\mathcal{N}^n, \mathcal{N}^{n-1}, \cdots, \mathcal{N}^1\}$.
  - The superscript number denotes the level of the IT which the node is at.
  - Observed plane feature: $\boldsymbol{\pi} = [\boldsymbol{n}^T, d]^T$.
  - Observed line feature: $\mathcal{L} = [\boldsymbol{u}^T, \boldsymbol{v}^T]^T$.

Problem Formulation

- Interpretation Tree (IT)
  - The range of possible pairings of features represented in two different coordinate systems can be cast in the form of an interpretation tree (IT). Each node of the IT has N_r possible descendants, each representing an interpretation in which $\mathcal{F}_{ci}$ is associated with a different feature in the reference frame. There are a total of N_c levels in the tree, level i indicating the possible pairings of $\mathcal{F}_{ci}$ with the feature in the reference frame.
  - An n-interpretation is any path from the root node to a node at level n in the IT; it is a list of n pairings of features.

Problem Formulation

- Interpretation Tree

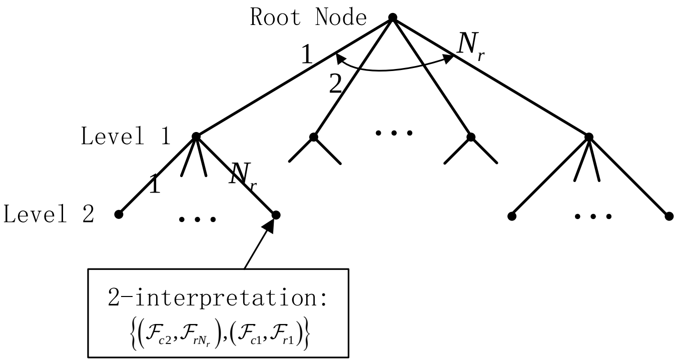


Fig. 1:

Inter-Node Consistency

- Definition 1: **inter-node consistency**
  - Given two nodes in the IT $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i)$ and $\mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$, if there exist (unique or multiple) rigid transformations $\boldsymbol{R} \in \mathbb{SO}(3), \boldsymbol{t} \in \mathbb{R}^3$ in 3D space such that

$$\begin{aligned} \mathcal{F}_c^i &= T(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}) \\ \mathcal{F}_c^j &= T(\mathcal{F}_r^j, \boldsymbol{R}, \boldsymbol{t}) \end{aligned} \tag{1}$$

with $T(\mathcal{F}, \boldsymbol{R}, \boldsymbol{t})$ denoting a 3D transform of feature $\mathcal{F}$ via $\boldsymbol{R}, \boldsymbol{t}$, then $\mathcal{N}^i$ and $\mathcal{N}^j$ are **inter-node consistent** under $\boldsymbol{R}, \boldsymbol{t}$.

Inter-Node Consistency

- Inter-Node Consistency
  - Rotation consistency.
  - Translation consistency.

Rotation consistency

- Rotation consistency
  - For geometric features, the 3D rigid rotation can be formulated as $\boldsymbol{e}_c = \boldsymbol{R}\boldsymbol{e}_r$, with $\boldsymbol{e}$ being the unit direction vector of the feature $\mathcal{F}$.

$$\boldsymbol{e} = \begin{cases} \boldsymbol{n} & \textit{plane feature} \\ \boldsymbol{v} & \textit{line feature} \end{cases} \tag{2}$$

  - Given two nodes in the IT $\mathcal{N}^i$ and $\mathcal{N}^j$, solve for the rotation $\boldsymbol{R} \in \mathbb{SO}(3)$ that satisfies

$$\boldsymbol{e}_c^i = \boldsymbol{R}\boldsymbol{e}_r^i \quad \boldsymbol{e}_c^j = \boldsymbol{R}\boldsymbol{e}_r^j \tag{3}$$

Rotation consistency

- Rotation consistency
  - We represent the rotation by

$$\boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}, \theta) = \cos\theta \boldsymbol{I} + (1 - \cos\theta)\boldsymbol{r}^T\boldsymbol{r} + \sin\theta[\boldsymbol{r}]_\times \tag{4}$$

where $\boldsymbol{r}$ is the axis about which the rotation takes place, and θ is the angle of rotation about that axis. Known that any rotation will preserve the angle between the transformed vector and the direction of rotation, the set of potential directions is given by

$$\left\{\boldsymbol{r}|\boldsymbol{r}^T\boldsymbol{e}_c^i = \boldsymbol{r}^T\boldsymbol{e}_r^i\right\} \cup \left\{\boldsymbol{r}|\boldsymbol{r}^T\boldsymbol{e}_c^j = \boldsymbol{r}^T\boldsymbol{e}_r^j\right\} \tag{5}$$

  - Given the axis of rotation $\boldsymbol{r}$ and a pair of feature directions $\boldsymbol{e}_c$ and $\boldsymbol{e}_r$, the rotation angle can be determined by $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c, \boldsymbol{e}_r) = \mathrm{atan2}(\sin\theta, \cos\theta)$.

$$\cos\theta = 1 - \frac{1 - \boldsymbol{e}_r^T\boldsymbol{e}_c}{1 - (\boldsymbol{r}^T\boldsymbol{e}_c)(\boldsymbol{r}^T\boldsymbol{e}_r)} \quad \sin\theta = \frac{(\boldsymbol{r} \times \boldsymbol{e}_r)^T\boldsymbol{e}_c}{1 - (\boldsymbol{r}^T\boldsymbol{e}_c)(\boldsymbol{r}^T\boldsymbol{e}_r)} \tag{6}$$

Rotation consistency

- According to the spatial configuration of the four feature directions, solution to Eq.(3) can be classified into four cases:
- **Case I**: $\boldsymbol{e}_c^i = \boldsymbol{e}_c^j \doteq \boldsymbol{e}_c$, $\boldsymbol{e}_r^i = \boldsymbol{e}_r^j \doteq \boldsymbol{e}_r$.
  - The rotation axis $\boldsymbol{r}$ satisfies $\boldsymbol{r}^T(\boldsymbol{e}_c - \boldsymbol{e}_r) = 0$ and $\boldsymbol{r}^T\boldsymbol{r} = 1$.
  - It can be computed by $\boldsymbol{r} = \boldsymbol{r}_x \cos\varphi + \boldsymbol{r}_y \sin\varphi$ and has one degree of freedom φ.

$$\boldsymbol{r}_x = \frac{\boldsymbol{e}_r \times \boldsymbol{e}_c}{\|\boldsymbol{e}_r \times \boldsymbol{e}_c\|} \quad \boldsymbol{r}_y = \frac{\boldsymbol{e}_r + \boldsymbol{e}_c}{\|\boldsymbol{e}_r + \boldsymbol{e}_c\|} \tag{7}$$

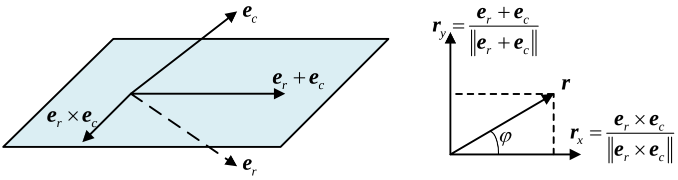


Fig. 2:

Rotation consistency

- **Case II**:
  (1) If $\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j$, $\boldsymbol{e}_r^i \neq \boldsymbol{e}_r^j$, $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$, then $\boldsymbol{R} = \boldsymbol{I}$.
  (2) If $\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j$, $\boldsymbol{e}_r^i \neq \boldsymbol{e}_r^j$, $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j$, then $\boldsymbol{r} = \boldsymbol{e}_c^i$, $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$.
  (3) If $\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j$, $\boldsymbol{e}_r^i \neq \boldsymbol{e}_r^j$, $\boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$, then $\boldsymbol{r} = \boldsymbol{e}_c^j$, $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$.

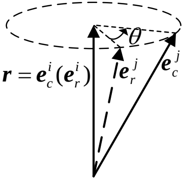


Fig. 3: Case II (2)

Rotation consistency

- **Case III**: $(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) = 0$ $(\boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j,\ \boldsymbol{e}_r^i \neq \boldsymbol{e}_r^j,\ \boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i,\ \boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j)$
  - Solve for $\boldsymbol{r}$ such that $\Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i) = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$. From (6)

$$\begin{aligned} 1 - \frac{1 - \boldsymbol{e}_r^{i\,T} \boldsymbol{e}_c^i}{1 - (\boldsymbol{r}^T \boldsymbol{e}_c^i)(\boldsymbol{r}^T \boldsymbol{e}_r^i)} &= 1 - \frac{1 - \boldsymbol{e}_r^{j\,T} \boldsymbol{e}_c^j}{1 - (\boldsymbol{r}^T \boldsymbol{e}_c^j)(\boldsymbol{r}^T \boldsymbol{e}_r^j)} \\ \frac{(\boldsymbol{r} \times \boldsymbol{e}_r^i)^T \boldsymbol{e}_c^i}{1 - (\boldsymbol{r}^T \boldsymbol{e}_c^i)(\boldsymbol{r}^T \boldsymbol{e}_r^i)} &= \frac{(\boldsymbol{r} \times \boldsymbol{e}_r^j)^T \boldsymbol{e}_c^j}{1 - (\boldsymbol{r}^T \boldsymbol{e}_c^j)(\boldsymbol{r}^T \boldsymbol{e}_r^j)} \end{aligned} \tag{8}$$

Rotation consistency

- **Case III**:
  - Denote

$$\alpha_i = \frac{1}{2} < \boldsymbol{e}_r^i, \boldsymbol{e}_c^i >, \quad \alpha_j = \frac{1}{2} < \boldsymbol{e}_r^j, \boldsymbol{e}_c^j >$$
$$\beta_i = < \boldsymbol{r}, \boldsymbol{e}_r^i > = < \boldsymbol{r}, \boldsymbol{e}_c^i >, \quad \beta_j = < \boldsymbol{r}, \boldsymbol{e}_r^j > = < \boldsymbol{r}, \boldsymbol{e}_c^j > \tag{9}$$
$$\gamma_i = < \boldsymbol{r}, \boldsymbol{e}_r^i \times \boldsymbol{e}_c^i >, \quad \gamma_j = < \boldsymbol{r}, \boldsymbol{e}_r^j \times \boldsymbol{e}_c^j >$$

where $< \cdot >$ means the included angle.

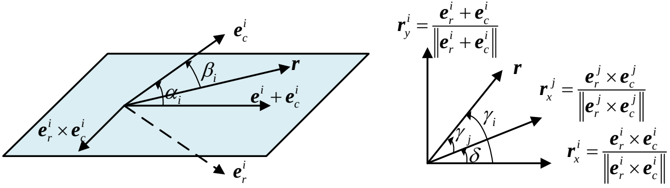


Fig. 4:

Rotation consistency

- **Case III**:
  - Eq.(8) can be rewritten as

$$\frac{\sin^2 \alpha_i}{\sin^2 \beta_i} = \frac{\sin^2 \alpha_j}{\sin^2 \beta_j} \tag{10}$$

$$\frac{\cos \gamma_i \sin \alpha_i \cos \alpha_i}{\sin^2 \beta_i} = \frac{\cos \gamma_j \sin \alpha_j \cos \alpha_j}{\sin^2 \beta_j} \tag{11}$$

  - Combining (10) and (11),

$$\frac{\cos \gamma_i}{\tan \alpha_i} = \frac{\cos \gamma_j}{\tan \alpha_j} \tag{12}$$

Rotation consistency

- **Case III**:
  - Let

$$\boldsymbol{r}_x^i = \frac{\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i}{\|\boldsymbol{e}_r^i \times \boldsymbol{e}_c^i\|} \quad \boldsymbol{r}_y^i = \frac{\boldsymbol{e}_r^i + \boldsymbol{e}_c^i}{\|\boldsymbol{e}_r^i + \boldsymbol{e}_c^i\|} \quad \boldsymbol{r}_x^j = \frac{\boldsymbol{e}_r^j \times \boldsymbol{e}_c^j}{\|\boldsymbol{e}_r^j \times \boldsymbol{e}_c^j\|} \quad \boldsymbol{r}_y^j = \frac{\boldsymbol{e}_r^j + \boldsymbol{e}_c^j}{\|\boldsymbol{e}_r^j + \boldsymbol{e}_c^j\|} \tag{13}$$

  - From (9) γ_i can be rewritten as

$$\gamma_i = \text{atan2}\left(\boldsymbol{r}^T \boldsymbol{r}_y^i, \boldsymbol{r}^T \boldsymbol{r}_x^i\right) \tag{14}$$

    And let

$$\delta = \text{atan2}\left({\boldsymbol{r}_x^j}^T \boldsymbol{r}_y^i, {\boldsymbol{r}_x^j}^T \boldsymbol{r}_x^i\right) \tag{15}$$

  - From (9) (14) and (15), we know that $\gamma_j = \gamma_i - \delta$.
  - Substitute γ_j in (12) and γ_i can be solved. Thus, the axis $\boldsymbol{r}$ can be computed by

$$\boldsymbol{r} = \boldsymbol{r}_x^i \cos\gamma_i + \boldsymbol{r}_y^i \sin\gamma_i. \tag{16}$$

Rotation consistency

- **Case IV** $((\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) \neq 0, \boldsymbol{e}_c^i \neq \boldsymbol{e}_c^j, \boldsymbol{e}_r^i \neq \boldsymbol{e}_r^j, \boldsymbol{e}_c^i \neq \boldsymbol{e}_r^i, \boldsymbol{e}_c^j \neq \boldsymbol{e}_r^j)$
  - The rotation axis can be determined by

$$\boldsymbol{r} = \eta(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) \tag{17}$$

    η is the normalizing factor.
  - Compute $\theta_i = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$ and $\theta_j = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$.
  - If $\theta_i = \theta_j$, then the two nodes satisfy the rotation consistency.

Algorithm 1 rotation consistency

Input: $\{\boldsymbol{e}_c^i, \boldsymbol{e}_r^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j\}$ – four directions of features from two nodes $\mathcal{N}^i, \mathcal{N}^j$.

Output: $\mathcal{R} = \{\boldsymbol{R} | \boldsymbol{e}_c^i = \boldsymbol{R}\boldsymbol{e}_r^i, \boldsymbol{e}_c^j = \boldsymbol{R}\boldsymbol{e}_r^j\}$

function ROTATIONCONSISTENCY($\boldsymbol{e}_c^i, \boldsymbol{e}_r^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j$)
    if $\boldsymbol{e}_c^i = \boldsymbol{e}_c^j$ and $\boldsymbol{e}_r^i = \boldsymbol{e}_r^j$ **then**
        $\boldsymbol{r}(\varphi) = \boldsymbol{r} = \boldsymbol{r}_x \cos\varphi + \boldsymbol{r}_y \sin\varphi, \varphi \in \mathbb{R}$, with $\boldsymbol{r}_x$ and $\boldsymbol{r}_y$ computed by (7).
        $\theta(\varphi) = \Theta(\boldsymbol{r}(\varphi), \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$.
        $\mathcal{R} = \{\boldsymbol{R} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}(\varphi), \theta(\varphi)), \varphi \in \mathbb{R}\}$.
    else if $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ and $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ **then**
        $\mathcal{R} = \{\boldsymbol{I}\}$.
    else if $\boldsymbol{e}_c^i = \boldsymbol{e}_r^i$ **then**
        $\boldsymbol{r} = \boldsymbol{e}_c^i$, $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$.
        $\mathcal{R} = \{\mathrm{Rot}(\boldsymbol{r}, \theta)\}$.
    else if $\boldsymbol{e}_c^j = \boldsymbol{e}_r^j$ **then**
        $\boldsymbol{r} = \boldsymbol{e}_c^j$, $\theta = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$.
        $\mathcal{R} = \{\mathrm{Rot}(\boldsymbol{r}, \theta)\}$.
    else if $(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j) = 0$ **then**
        $\boldsymbol{r} = \boldsymbol{r}_x^i \cos\gamma_i + \boldsymbol{r}_y^i \sin\gamma_i$, $\theta = 0.5\left(\Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i) + \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)\right)$.

        $\mathcal{R} = \{\text{Rot}(\boldsymbol{r}, \theta)\}$.
    else
        $\boldsymbol{r} = \eta(\boldsymbol{e}_c^i - \boldsymbol{e}_r^i) \times (\boldsymbol{e}_c^j - \boldsymbol{e}_r^j)$, $\theta_i = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^i, \boldsymbol{e}_r^i)$ and $\theta_j = \Theta(\boldsymbol{r}, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j)$.
        if $\theta_i = \theta_j$ **then**
            $\mathcal{R} = \{\text{Rot}(\boldsymbol{r}, 0.5(\theta_i + \theta_j))\}$.
        else
            $\mathcal{R} = \emptyset$.
        end if
    end if
    return $\mathcal{R}$.
end function

Translation consistency

- Given two nodes in the IT $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i)$, $\mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$ and the set of consistent rotation $\mathcal{R}$, solve for the translation $\boldsymbol{t} \in \mathbb{R}^3$ such that $\mathcal{N}^i$ and $\mathcal{N}^j$ are inter-node consistent under $\boldsymbol{R}, \boldsymbol{t}$.
- Three different cases:
  - Plane-Plane case: $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$
  - Line-Line case: $\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$
  - Plane-Line case: $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$

Translation consistency

- **Plane-Plane case**: $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$
  - Solve the linear equations

$$\begin{aligned} {\boldsymbol{n}_c^i}^T \boldsymbol{t} &= d_r^i - d_c^i \\ {\boldsymbol{n}_c^j}^T \boldsymbol{t} &= d_r^j - d_c^j \end{aligned} \tag{18}$$

  - Let

$$\boldsymbol{A}\boldsymbol{t} = \boldsymbol{b} \tag{19}$$

$$\boldsymbol{A} = \begin{bmatrix} {\boldsymbol{n}_c^i}^T \\ {\boldsymbol{n}_c^j}^T \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} d_r^i - d_c^i \\ d_r^j - d_c^j \end{bmatrix} \tag{20}$$

Translation consistency

- **Plane-Plane case**:
  (1) If $\boldsymbol{n}_c^i = \boldsymbol{n}_c^j \doteq \boldsymbol{n}_c$ (parallel planes), $\mathrm{rank}(\boldsymbol{A}) = 1$. So it requires that $\mathrm{rank}([\boldsymbol{A}|\boldsymbol{b}]) = 1$, i.e., $d_r^i - d_c^i = d_r^j - d_c^j$. The solution $\boldsymbol{t}$ has two degrees of freedom.

$$\boldsymbol{t}(\boldsymbol{\mu}) = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \tag{21}$$

where

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{n}_c; \\ \boldsymbol{\mu} &\in \mathbb{R}^3; \\ \boldsymbol{t}_0 &= (\boldsymbol{A}'^T \boldsymbol{A}')^{-1} \boldsymbol{A}'^T \boldsymbol{b}' \\ \boldsymbol{A}' &= \begin{bmatrix} \boldsymbol{A} \\ [\boldsymbol{w}]_\times \end{bmatrix} \\ \boldsymbol{b}' &= \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \end{aligned} \tag{22}$$

Translation consistency

- **Plane-Plane case**:
  (2) If $\boldsymbol{n}_c^i \neq \boldsymbol{n}_c^j$ (non-parallel planes), $\text{rank}(\boldsymbol{A}) = 2$. $\text{rank}([\boldsymbol{A}|\boldsymbol{b}]) = 2$ holds true. The solution $\boldsymbol{t}$ has one degree of freedom.

$$\boldsymbol{t}(\mu) = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \tag{23}$$

where

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{n}_c^i \times \boldsymbol{n}_c^j; \\ \mu &\in \mathbb{R}; \\ \boldsymbol{t}_0 &= (\boldsymbol{A}'^T \boldsymbol{A}')^{-1} \boldsymbol{A}'^T \boldsymbol{b}' \\ \boldsymbol{A}' &= \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{w}^T \end{bmatrix} \\ \boldsymbol{b}' &= \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix} \end{aligned} \tag{24}$$

Translation consistency

- **Line-Line case**: $\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$
  - Solve the linear equations

$$\begin{aligned} [\boldsymbol{v}_c^i]_\times \boldsymbol{t} &= \boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i \\ [\boldsymbol{v}_c^j]_\times \boldsymbol{t} &= \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \end{aligned} \tag{25}$$

  - Let

$$\boldsymbol{A}\boldsymbol{t} = \boldsymbol{b} \tag{26}$$

$$\boldsymbol{A} = \begin{bmatrix} [\boldsymbol{v}_c^i]_\times \\ [\boldsymbol{v}_c^j]_\times \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i \\ \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \end{bmatrix} \tag{27}$$

Translation consistency

- **Line-Line case**:
  (1) If $\boldsymbol{v}_c^i = \boldsymbol{v}_c^j \doteq \boldsymbol{v}_c$ (parallel lines), rank$(\boldsymbol{A}) = 2$. If rank$([\boldsymbol{A}|\boldsymbol{b}]) = 2$, it requires that

$$\boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i = \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \tag{28}$$

Eq.(28) is equivalent to (29) and (30).

$$\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\| = \|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\| \tag{29}$$

$$\boldsymbol{R}\frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\|} = \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\|} \tag{30}$$

Translation consistency

- **Line-Line case**:

  As known from Algorithm 1, if $\boldsymbol{v}_c^i = \boldsymbol{v}_c^j \doteq \boldsymbol{v}_c$, it is true that $\boldsymbol{v}_r^i = \boldsymbol{v}_r^j \doteq \boldsymbol{v}_r$ and the consistent rotation $\boldsymbol{R}$ has one degree of freedom. So we need to solve for a rotation to satisfy (30) and $\boldsymbol{v}_c = \boldsymbol{R}_r$ simultaneously using the function defined in Algorithm 1.

$$\textsc{RotationConsistency}(\boldsymbol{v}_c, \boldsymbol{v}_r, \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\|}, \frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\|})$$

Translation consistency

- **Line-Line case**:

  If the returning value is not Ø and (29) is satisfied, then the two nodes are translation consistent, and the resultant translation has one degree of freedom.

$$\boldsymbol{t}(\mu) = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \tag{31}$$

where

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{v}_c; \\ \mu &\in \mathbb{R}; \\ \boldsymbol{t}_0 &= (\boldsymbol{A}'^T \boldsymbol{A}')^{-1} \boldsymbol{A}'^T \boldsymbol{b}' \\ \boldsymbol{A}' &= \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{w}^T \end{bmatrix} \\ \boldsymbol{b}' &= \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix} \end{aligned} \tag{32}$$

Translation consistency

- **Line-Line case**:
  (2) If $\boldsymbol{v}_c^i \neq \boldsymbol{v}_c^j$ (non-parallel lines), $\text{rank}(\boldsymbol{A}) = 3$. Then $\text{rank}([\boldsymbol{A}|\boldsymbol{b}]) = 3$ requires that

$$l(\mathcal{L}_r^i, \mathcal{L}_r^j) - l(\mathcal{L}_c^i, \mathcal{L}_c^j) = (\boldsymbol{R}\boldsymbol{u}_r^i - \boldsymbol{u}_c^i)^T \boldsymbol{v}_c^j + (\boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j)^T \boldsymbol{v}_c^i = 0 \qquad (33)$$

  where $l(\mathcal{L}^i, \mathcal{L}^j)$ represents the vertical distance between two 3D lines. In this case the translation is constrained and can be computed by

$$\boldsymbol{t} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \qquad (34)$$

Translation consistency

- **Plane-Line case**: $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$
  - Solve the linear equations

$$\begin{aligned} {\boldsymbol{n}_c^i}^T \boldsymbol{t} &= d_r^i - d_c^i \\ [\boldsymbol{v}_c^j]_\times \boldsymbol{t} &= \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \end{aligned} \tag{35}$$

  - Let

$$\boldsymbol{A}\boldsymbol{t} = \boldsymbol{b} \tag{36}$$

$$\boldsymbol{A} = \begin{bmatrix} {\boldsymbol{n}_c^i}^T \\ [\boldsymbol{v}_c^j]_\times \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} d_r^i - d_c^i \\ \boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j \end{bmatrix} \tag{37}$$

Translation consistency

- **Plane-Line case**:
  (1) If ${\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j = 0$ (the plane and the line are parallel), $\mathrm{rank}(\boldsymbol{A}) = 2$. Then $\mathrm{rank}([\boldsymbol{A}|\boldsymbol{b}]) = 2$ requires that

$$l(\pi_r^i, \mathcal{L}_r^j) - l(\pi_c^i, \mathcal{L}_c^j) = {\boldsymbol{n}_c^i}^T [\boldsymbol{v}_c^j]_\times (\boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j) + (d_r^i - d_c^i) = 0 \tag{38}$$

  where $l(\pi^i, \mathcal{L}^j)$ represents the vertical distance between a plane and a line which are parallel to each other.

Translation consistency

- **Plane-Line case**:

  The resultant translation has one degree of freedom.

$$\boldsymbol{t}(\mu) = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \tag{39}$$

where

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{v}_c^j; \\ \mu &\in \mathbb{R}; \\ \boldsymbol{t}_0 &= (\boldsymbol{A}'^T \boldsymbol{A}')^{-1} \boldsymbol{A}'^T \boldsymbol{b}' \\ \boldsymbol{A}' &= \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{w}^T \end{bmatrix} \\ \boldsymbol{b}' &= \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix} \end{aligned} \tag{40}$$

Translation consistency

- **Plane-Line case**:
  (2) If $\boldsymbol{n}_c^i = \boldsymbol{v}_c^j$ (the line is vertical to the plane), $\text{rank}(\boldsymbol{A}) = 3$. However, in this case, the consistent rotation has one degree of freedom.
  Solving (36) directly yields

$$\begin{aligned} \boldsymbol{t}(\varphi) &= (d_r^i - d_c^i)\boldsymbol{n}_c^i - [\boldsymbol{v}_c^j]_\times(\boldsymbol{R}\boldsymbol{u}_r^j - \boldsymbol{u}_c^j) \\ &= (d_r^i - d_c^i)\boldsymbol{n}_c^i + \boldsymbol{v}_c^j \times \boldsymbol{u}_c^j + \boldsymbol{R}(\boldsymbol{u}_r^j \times \boldsymbol{v}_r^j) \\ &\doteq \boldsymbol{t}_0 + \boldsymbol{R}\boldsymbol{w}, \boldsymbol{R} \in \mathcal{R}. \end{aligned} \tag{41}$$

  The resultant translation has one degree of freedom φ as the rotation.

Translation consistency

- **Plane-Line case**:

  (3) If ${\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j \neq 0$ and $\boldsymbol{n}_c^i \neq \boldsymbol{v}_c^j$, $\text{rank}(\boldsymbol{A}) = 3$. In this case, $\text{rank}([\boldsymbol{A}|\boldsymbol{b}]) = 3$ holds true. So the translation is constrained and can be computed by

$$\boldsymbol{t} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{42}$$

Algorithm 2 translation consistency

Input: Two nodes in the IT $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i)$, $\mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$ and the set of consistent rotation $\mathcal{R}$.

Output: $\mathcal{S} = \mathcal{T} \cup \mathcal{R}' = \{\boldsymbol{t}, \boldsymbol{R}' | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}', \boldsymbol{t}), \mathcal{F}_c^j = T(\mathcal{F}_r^j, \boldsymbol{R}', \boldsymbol{t})\}$

function TranslationConsistency($\{\mathcal{N}^i, \mathcal{N}^j\}, \mathcal{R}$)
    $\mathcal{R}' = \mathcal{R}$.
    if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$ **then**
        if $\boldsymbol{n}_c^i = \boldsymbol{n}_c^j \doteq \boldsymbol{n}_c$ **then**
            if $d_r^i - d_c^i = d_r^j - d_c^j$ **then**
                $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$ (ref.(21)).
            else
                $\mathcal{T} = \emptyset$.
            end if
        else if $\boldsymbol{n}_c^i \neq \boldsymbol{n}_c^j$ **then**
            $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$ (ref.(23)).
        end if
    else if $\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ **then**
        if $\boldsymbol{v}_c^i = \boldsymbol{v}_c^j \doteq \boldsymbol{v}_c$ **then**

if $\left| \begin{array}{l} \|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\| = \|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\| \;\&\& \\ \textsc{RotationConsistency}(\boldsymbol{v}_c, \boldsymbol{v}_r, \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\|}, \frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\|}, \mathcal{R}) \neq \emptyset \end{array} \right.$ **then**

  $\mathcal{R}' = \textsc{RotationConsistency}(\boldsymbol{v}_c, \boldsymbol{v}_r, \frac{\boldsymbol{u}_c^i - \boldsymbol{u}_c^j}{\|\boldsymbol{u}_c^i - \boldsymbol{u}_c^j\|}, \frac{\boldsymbol{u}_r^i - \boldsymbol{u}_r^j}{\|\boldsymbol{u}_r^i - \boldsymbol{u}_r^j\|})$.

  $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$ (ref.(31)).

else

  $\mathcal{T} = \emptyset$.

end if

else if $\boldsymbol{v}_c^i \neq \boldsymbol{v}_c^j$ **then**

  if $l(\mathcal{L}_r^i, \mathcal{L}_r^j) - l(\mathcal{L}_c^i, \mathcal{L}_c^j) = 0$ **then**

    $\mathcal{T} = \{(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}\}$ (ref.(34)).

  else

    $\mathcal{T} = \emptyset$.

  end if

end if

else if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ **then**

  if ${\boldsymbol{n}_c^i}^T \boldsymbol{v}_c^j = 0$ **then**

    if $l(\pi_r^i, \mathcal{L}_r^j) - l(\pi_c^i, \mathcal{L}_c^j)$ **then**

        $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$ (ref.(39)).
      else
        $\mathcal{T} = \emptyset$.
      end if
    else if $\boldsymbol{n}_c^i = \boldsymbol{v}_c^j$ **then**
      $\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}\boldsymbol{w}, \boldsymbol{R} \in \mathcal{R}\}$ (ref.(41)).
    else
      $\mathcal{T} = \{(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}\}$ (ref.(42)).
    end if
  end if
  return $\mathcal{S} = \mathcal{T} \cup \mathcal{R}'$.
end function

Algorithm 3 inter-node consistency

Input: Two nodes in the IT $\mathcal{N}^i = (\mathcal{F}_c^i, \mathcal{F}_r^i)$, $\mathcal{N}^j = (\mathcal{F}_c^j, \mathcal{F}_r^j)$.

Output: $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}), \mathcal{F}_c^j = T(\mathcal{F}_r^j, \boldsymbol{R}, \boldsymbol{t})\}$

function INTERNODECONSISTENCY($\{\mathcal{N}^i, \mathcal{N}^j\}$)
  if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \pi_c^j, \mathcal{F}_r^j = \pi_r^j$ **then**
    $\boldsymbol{e}_c^i = \boldsymbol{n}_c^i, \boldsymbol{e}_r^i = \boldsymbol{n}_r^i, \boldsymbol{e}_c^j = \boldsymbol{n}_c^j, \boldsymbol{e}_r^j = \boldsymbol{n}_r^j$.
  else if $\mathcal{F}_c^i = \mathcal{L}_c^i, \mathcal{F}_r^i = \mathcal{L}_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ **then**
    $\boldsymbol{e}_c^i = \boldsymbol{v}_c^i, \boldsymbol{e}_r^i = \boldsymbol{v}_r^i, \boldsymbol{e}_c^j = \boldsymbol{v}_c^j, \boldsymbol{e}_r^j = \boldsymbol{v}_r^j$.
  else if $\mathcal{F}_c^i = \pi_c^i, \mathcal{F}_r^i = \pi_r^i, \mathcal{F}_c^j = \mathcal{L}_c^j, \mathcal{F}_r^j = \mathcal{L}_r^j$ **then**
    $\boldsymbol{e}_c^i = \boldsymbol{n}_c^i, \boldsymbol{e}_r^i = \boldsymbol{n}_r^i, \boldsymbol{e}_c^j = \boldsymbol{v}_c^j, \boldsymbol{e}_r^j = \boldsymbol{v}_r^j$.
  end if
  $\mathcal{R} =$ROTATIONCONSISTENCY($\boldsymbol{e}_c^i, \boldsymbol{e}_r^i, \boldsymbol{e}_c^j, \boldsymbol{e}_r^j$).
  $\mathcal{S} =$TRANSLATIONCONSISTENCY($\{\mathcal{N}^i, \mathcal{N}^j\}, \mathcal{R}$).
  return $\mathcal{S}$.
end function

Incrementally Constructed Consistent Transform Model

- Definition 2: incrementally constructed **consistent transform model**
  - For an n-interpretation $\mathcal{P}_n = \{\mathcal{N}^n, \mathcal{N}^{n-1}, \cdots, \mathcal{N}^1\}$, a consistent transform model is defined by
  $$\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n) \tag{43}$$
  with
  $$\mathcal{S}_n = \{\boldsymbol{R}, \boldsymbol{t} | \mathcal{F}_c^i = T(\mathcal{F}_r^i, \boldsymbol{R}, \boldsymbol{t}), \forall i = 1, \cdots, n\}. \tag{44}$$
  - When a new node $\mathcal{N}^{n+1}$ is added to the interpretation, the resultant transform model $\mathcal{S}^{n+1}$ can be incrementally updated. $\mathcal{P}_{n+1} = \{\mathcal{N}^{n+1}, \mathcal{N}^n, \cdots, \mathcal{N}^1\}$ is simply a union of $\mathcal{P}_n$ and $\mathcal{N}^{n+1}$. The updating method of $\mathcal{S}_n$ will be detailed in the following.

Transform Model Updating

- As seen from the output of the Algorithm 3, the solution set has four different forms.
  (1) 1 rotDoF[1] & 2 transDoF[2]:

$$\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}(\varphi), \theta(\varphi)), \varphi \in \mathbb{R}, \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$$

  (2) 1 rotDoF: $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t} | \boldsymbol{R} = \mathrm{Rot}(\boldsymbol{r}(\varphi), \theta(\varphi)), \varphi \in \mathbb{R}, \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}\boldsymbol{w}\}$
  (3) 1 transDoF: $\mathcal{S} = \{\boldsymbol{R}\} \cup \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$
  (4) constrained: $\mathcal{S} = \{\boldsymbol{R}, \boldsymbol{t}\}$
- A new node $\mathcal{N}^{n+1}$ is added to $\mathcal{P}_n$ yielding $\mathcal{P}_{n+1}$.
  The consistent transform model $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$ is updated by fusing $\textsc{InterNodeConsistency}(\{\mathcal{N}^{n+1}, \mathcal{N}^k\}), k = 1, \cdots, n$ into $\mathcal{S}_n$ yielding an updated solution set for the transform $\mathcal{S}_{n+1}$ to make the $(n+1)$-interpretation $\mathcal{P}_{n+1}$ consistent.
- To this end, a function $\mathcal{S}'' = \textsc{Fuse}(\mathcal{S}, \mathcal{S}')$ is defined to fuse two different set of transforms into an updated one $\mathcal{S}''$.

[1] rotDoF: rotational degree of freedom.
[2] transDoF: translational degree of freedom.

Transform Model Updating

(1) $\mathcal{S}$ has 1 rotDoF & 2 transDoF

(a) $\mathcal{S}'$ has 1 rotDoF & 2 transDoF
As known from Algorithm 1, if the consistent rotation has 1 DoF, the features in the same coordinate system are parallel to each other, i.e., they share the same feature direction. We denote them by $\boldsymbol{e}_c, \boldsymbol{e}_r, \boldsymbol{e}'_c, \boldsymbol{e}'_r$. Then the set of consistent rotation $\mathcal{R}''$ in $\mathcal{S}''$ can be determined by

$$\mathcal{R}'' = \textsc{RotationConsistency}(\boldsymbol{e}_c, \boldsymbol{e}_r, \boldsymbol{e}'_c, \boldsymbol{e}'_r).$$

Transform Model Updating

(1) $\mathcal{S}$ has 1 rotDoF & 2 transDoF

(a) $\mathcal{S}'$ has 1 rotDoF & 2 transDoF
The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented respectively by

$$\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}, \quad \mathcal{T}' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}'_0 + [\boldsymbol{w}']_\times \boldsymbol{\mu}', \boldsymbol{\mu}' \in \mathbb{R}^3\}.$$

Let $\Delta \boldsymbol{t}_0 = \boldsymbol{t}_0 - \boldsymbol{t}'_0$.

(i) If $\|\Delta \boldsymbol{t}_0\| = 0$ and $\boldsymbol{w} \times \boldsymbol{w}' = 0$, then $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$.
(ii) If $\|\Delta \boldsymbol{t}_0\| = 0$ and $\boldsymbol{w} \times \boldsymbol{w}' \neq 0$, then $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu(\boldsymbol{w} \times \boldsymbol{w}'), \mu \in \mathbb{R}\}$.
(iii) If $\|\Delta \boldsymbol{t}_0\| \neq 0$, $\boldsymbol{w} \times \boldsymbol{w}' = 0$ and $\Delta \boldsymbol{t}_0^T \boldsymbol{w} = 0$, $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}$.
(iv) If $\|\Delta \boldsymbol{t}_0\| \neq 0$ and $\boldsymbol{w} \times \boldsymbol{w}' \neq 0$, rank$([-[\boldsymbol{w}]_\times, [\boldsymbol{w}']_\times]) = 3$. Solve

$$\begin{bmatrix} -[\boldsymbol{w}]_\times & [\boldsymbol{w}']_\times \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}' \end{bmatrix} = \Delta \boldsymbol{t}_0.$$

Then, $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = (\boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}) + \mu(\boldsymbol{w} \times \boldsymbol{w}'), \mu \in \mathbb{R}\}$.

Thus, the returning value of the function is $\mathcal{S}'' = \mathcal{R}'' \cup \mathcal{T}''$.

Transform Model Updating

(1) $\mathcal{S}$ has 1 rotDoF & 2 transDoF

(b) $\mathcal{S}'$ has 1 rotDoF

The determination of $\mathcal{R}''$ is the same as (a).

The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented respectively by

$$\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}, \quad \mathcal{T}' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}'_0 + \boldsymbol{R}'\boldsymbol{w}', \boldsymbol{R}' \in \mathcal{R}''\}.$$

(i) If $\mathcal{R}''$ has 1DoF and $\boldsymbol{t}_0^T \boldsymbol{w} = \boldsymbol{t}'^T_0 \boldsymbol{w}$, $\mathcal{T}'' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}'_0 + \boldsymbol{R}'\boldsymbol{w}', \boldsymbol{R}' \in \mathcal{R}''\}$.

(ii) If $\mathcal{R}'' = \{\boldsymbol{R}''\}$ is constrained, then $\mathcal{T}'$ is also constrained

$\mathcal{T}' = \{\boldsymbol{t}'\}, \boldsymbol{t}' = \boldsymbol{t}'_0 + \boldsymbol{R}''\boldsymbol{w}'$.

And let $\Delta \boldsymbol{t} = \boldsymbol{t}_0 - \boldsymbol{t}'$.

If $\|\Delta \boldsymbol{t}\| = 0$, then $\mathcal{T}'' = \{\boldsymbol{t}'\}$.

If $\|\Delta \boldsymbol{t}\| \neq 0$ and $\Delta \boldsymbol{t}^T \boldsymbol{w} = 0$, then $\mathcal{T}'' = \{\boldsymbol{t}'\}$.

Transform Model Updating

(1) $\mathcal{S}$ has 1 rotDoF & 2 transDoF

(c) $\mathcal{S}'$ has 1 transDoF

The determination of $\mathcal{R}''$ is the same as (a).

The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented respectively by

$$\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^3\}, \quad \mathcal{T}' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}'_0 + \mu' \boldsymbol{w}', \mu' \in \mathbb{R}\}.$$

Let $\Delta \boldsymbol{t}_0 = \boldsymbol{t}_0 - \boldsymbol{t}'_0$.

(i) If $\|\Delta \boldsymbol{t}_0\| = 0$ and $\boldsymbol{w}^T \boldsymbol{w}' = 0$, then $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}'_0 + \mu' \boldsymbol{w}', \mu' \in \mathbb{R}\}$.

(ii) If $\|\Delta \boldsymbol{t}_0\| = 0$ and $\boldsymbol{w}^T \boldsymbol{w}' \neq 0$, then $\mathcal{T}'' = \{\boldsymbol{t}_0\}$.

(iii) If $\|\Delta \boldsymbol{t}_0\| \neq 0$, $\boldsymbol{w}^T \boldsymbol{w}' = 0$ and $\Delta \boldsymbol{t}_0^T \boldsymbol{w} = 0$, then $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}'_0 + \mu' \boldsymbol{w}', \mu' \in \mathbb{R}\}$.

(iv) If $\|\Delta \boldsymbol{t}_0\| \neq 0$ and $\boldsymbol{w}^T \boldsymbol{w}' \neq 0$, $\text{rank}([-[\boldsymbol{w}]_\times, \boldsymbol{w}']) = 3$. Solve

$$\begin{bmatrix} -[\boldsymbol{w}]_\times & \boldsymbol{w}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu} \\ \mu' \end{bmatrix} = \Delta \boldsymbol{t}_0.$$

Then, $\mathcal{T}'' = \{\boldsymbol{t}_0 + [\boldsymbol{w}]_\times \boldsymbol{\mu}\}$.

(d) $\mathcal{S}'$ is constrained

Then the solution is the same as (b)(ii).

Transform Model Updating

(2) $\mathcal{S}$ has 1 rotDoF

(a) $\mathcal{S}'$ has 1 1 rotDoF & 2 transDoF [same as (1)(b)]

(b) $\mathcal{S}'$ has 1 rotDoF

The determination of $\mathcal{R}''$ is the same as (1)(a).

(i) If $\mathcal{R}''$ has 1DoF, The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented respectively by

$$\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}\boldsymbol{w}, \boldsymbol{R} \in \mathcal{R}''\}, \quad \mathcal{T}' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}_0' + \boldsymbol{R}'\boldsymbol{w}', \boldsymbol{R}' \in \mathcal{R}''\}.$$

In this case, the features in the same coordinate system share the same feature direction, which are denoted by $\boldsymbol{e}_c, \boldsymbol{e}_r$. If $\|\boldsymbol{w} - \boldsymbol{w}'\| = \|\boldsymbol{t}_0 - \boldsymbol{t}_0'\|$,

$$\mathcal{R}'' = \textsc{RotationConsistency}(\boldsymbol{e}_c, \boldsymbol{e}_r, \frac{\boldsymbol{w} - \boldsymbol{w}'}{\|\boldsymbol{w} - \boldsymbol{w}'\|}, \frac{\boldsymbol{t}_0 - \boldsymbol{t}_0'}{\|\boldsymbol{t}_0 - \boldsymbol{t}_0'\|}).$$

(ii) If $\mathcal{R}'' = \{\boldsymbol{R}''\}$ is constrained, then

$$\mathcal{T} == \{\boldsymbol{t}\}, \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}''\boldsymbol{w}, \quad \mathcal{T}' = \{\boldsymbol{t}'\}, \boldsymbol{t}' = \boldsymbol{t}_0' + \boldsymbol{R}''\boldsymbol{w}'.$$

If $\|\boldsymbol{t}\| = \|\boldsymbol{t}'\|$, then $\mathcal{T}'' = \{\boldsymbol{t}\}$.

Transform Model Updating

(2) $\mathcal{S}$ has 1 rotDoF

(c) $\mathcal{S}'$ has 1 transDoF

(i) The sets of consistent rotation in $\mathcal{S}$ and $\mathcal{S}'$ are represented by $\mathcal{R} = \{\boldsymbol{R} | R = \mathrm{Rot}(\boldsymbol{r}(\varphi), \theta(\varphi))\}$ and $\mathcal{R}' = \{\boldsymbol{R}'\}, \boldsymbol{R}' = \mathrm{Rot}(\boldsymbol{r}', \theta')$, respectively.
The directions of the features that generates $\mathcal{S}$ in two different coordinate systems are denoted by $\boldsymbol{e}_c$ and $\boldsymbol{e}_r$, respectively. If $(\boldsymbol{e}_c - \boldsymbol{e}_r)^T \boldsymbol{r}' = 0$, compute $\theta = \Theta(\boldsymbol{r}', \boldsymbol{e}_c, \boldsymbol{e}_r)$.
Thus, if $\theta = \theta'$, then $\mathcal{R}'' = \{\boldsymbol{R}'\}$.

(ii) The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented by $\mathcal{T} = \{\boldsymbol{t}\}, \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}'\boldsymbol{w}$ and $\mathcal{T}' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}'_0 + \mu' \boldsymbol{w}', \mu' \in \mathbb{R}\}$, respectively.
Let $\Delta \boldsymbol{t} = \boldsymbol{t} - \boldsymbol{t}'_0$.
If $\|\Delta \boldsymbol{t}\| = 0$, or $\|\Delta \boldsymbol{t}\| \neq 0$ && $\Delta \boldsymbol{t} \times \boldsymbol{w}' = 0$, then $\mathcal{T}'' = \mathcal{T}$.

(d) $\mathcal{S}'$ is constrained

(i) The determination of $\mathcal{R}''$ is the same as (2)(b)(i).

(ii) The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented by $\mathcal{T} = \{\boldsymbol{t}\}, \boldsymbol{t} = \boldsymbol{t}_0 + \boldsymbol{R}'\boldsymbol{w}$ and $\mathcal{T}' = \{\boldsymbol{t}'\}$, respectively.
If $\|\boldsymbol{t} - \boldsymbol{t}'\| = 0$, then $\mathcal{T}'' = \mathcal{T}$.

Transform Model Updating

(3) $\mathcal{S}$ has 1 transDoF

(a) $\mathcal{S}'$ has 1 1 rotDoF & 2 transDoF [same as (1)(c)]

(b) $\mathcal{S}'$ has 1 rotDoF [same as (2)(c)]

(c) $\mathcal{S}'$ has 1 transDoF

The sets of consistent rotation in $\mathcal{S}$ and $\mathcal{S}'$ are $\mathcal{R} = \{\boldsymbol{R}\}$ and $\mathcal{R}' = \{\boldsymbol{R}'\}$.

If $\boldsymbol{R} = \boldsymbol{R}'$, then $\mathcal{R}'' = \mathcal{R}$.

The sets of consistent translation in $\mathcal{S}$ and $\mathcal{S}'$ are represented respectively by

$$\mathcal{T} = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}, \quad \mathcal{T}' = \{\boldsymbol{t}' | \boldsymbol{t}' = \boldsymbol{t}'_0 + \mu' \boldsymbol{w}', \mu' \in \mathbb{R}\}.$$

Let $\Delta \boldsymbol{t}_0 = \boldsymbol{t}_0 - \boldsymbol{t}'_0$.

(i) If $\|\Delta \boldsymbol{t}_0\| = 0$ and $\boldsymbol{w} \times \boldsymbol{w}' = 0$, then $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$.

(ii) If $\|\Delta \boldsymbol{t}_0\| = 0$ and $\boldsymbol{w} \times \boldsymbol{w}' \neq 0$, then $\mathcal{T}'' = \{\boldsymbol{t}_0\}$.

(iii) If $\|\Delta \boldsymbol{t}_0\| \neq 0$, $\boldsymbol{w} \times \boldsymbol{w}' = 0$ and $\Delta \boldsymbol{t}_0 \times \boldsymbol{w} = 0$, then $\mathcal{T}'' = \{\boldsymbol{t} | \boldsymbol{t} = \boldsymbol{t}_0 + \mu \boldsymbol{w}, \mu \in \mathbb{R}\}$.

(iv) If $\|\Delta \boldsymbol{t}_0\| \neq 0$, $\boldsymbol{w} \times \boldsymbol{w}' \neq 0$ and $\Delta \boldsymbol{t}_0^T (\boldsymbol{w} \times \boldsymbol{w}') = 0$, then $\Delta \boldsymbol{t}_0 \in \text{Columnspace}([-\boldsymbol{w}, \boldsymbol{w}'])$. Thus, the equation (45) has unique solution $[\mu, \mu']^T$.

$$\begin{bmatrix} -\boldsymbol{w} & \boldsymbol{w}' \end{bmatrix} \begin{bmatrix} \mu \\ \mu' \end{bmatrix} = \Delta \boldsymbol{t}_0. \tag{45}$$

Then, $\mathcal{T}'' = \{\boldsymbol{t}_0 + \mu \boldsymbol{w}\}$.

Transform Model Updating

(3) $\mathcal{S}$ has 1 transDoF
   (d) $\mathcal{S}'$ is constrained
   The determination of $\mathcal{R}''$ is the same as (3)(a). And $\mathcal{T}''$ is the same as (2)(b)(ii).

(4) $\mathcal{S}$ is constrained
   (a) $\mathcal{S}'$ has 1 1 rotDoF & 2 transDoF [same as (1)(d)]
   (b) $\mathcal{S}'$ has 1 rotDoF [same as (2)(d)]
   (c) $\mathcal{S}'$ has 1 transDoF [same as (3)(d)]
   (d) $\mathcal{S}'$ is constrained
   The determination of $\mathcal{R}''$ is the same as (3)(a). And $\mathcal{T}''$ is the same as (2)(c)(ii).

Transform Model Updating

- Until here the function $\mathcal{S}'' = \textsc{Fuse}(\mathcal{S}, \mathcal{S}')$ is defined.
- The output of the function can also be formulated into the four forms as the output of Algorithm 3.
- The transform model $\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n)$ can be updated to $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$ via Algorithm 4.

Algorithm 4 Consistent Transform Model Update

Input: The consistent transform model $\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n)$ for the n-interpretation $\mathcal{P}_n$, a newly added node $\mathcal{N}^{n+1}$.

Output: The updated consistent transform model $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$ for the (n+1)-interpretation $\mathcal{P}_{n+1}$.

function UpdateTransformModel($\mathcal{M}(\mathcal{P}_n, \mathcal{S}_n)$,$\mathcal{N}^{n+1}$)
    $\mathcal{P}_{n+1} = \{\mathcal{S}_{n+1}, \mathcal{P}_n\}$.
    $\mathcal{S} = \mathcal{S}_n$.
    for $k = 1$ to n **do**
        $\mathcal{S}' =$InterNodeConsistency($\{\mathcal{N}^n, \mathcal{N}^k\}$).
        $\mathcal{S} =$Fuse($\{\mathcal{S}, \mathcal{S}'\}$).
    end for
    $\mathcal{S}_{n+1} = \mathcal{S}$.
    return $\mathcal{M}(\mathcal{P}_{n+1}, \mathcal{S}_{n+1})$.
end function

Parameterization for Planes

- A plane equation in 3-space $\Pi_1 X + \Pi_2 Y + \Pi_3 Z + \Pi_4 = 0$ is unaffected by multiplication by a non-zero scalar – a plane has 3 degrees of freedom in 3-space.
- The homogeneous representation of the plane is $\mathbf{\Pi} = [\Pi_1, \Pi_2, \Pi_3, \Pi_4]^T \in \mathbb{P}^3$ in projective space. [3]
- Spherically normalizing the homogeneous vector $\mathbf{\Pi}$ yields $\boldsymbol{\pi} = \mathbf{\Pi} / \|\mathbf{\Pi}\| \in \mathbb{S}^3$.
- $\mathbb{S}^3$ is the 3-sphere in the space $\mathbb{R}^4$, which is a Lie group under the operation of quaternion multiplication when its elements are viewed as unit quaternions.

[3] R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”. Cambridge University Press, 2003, second Edition.

Minimal Parameterization for Planes

- Since there are four directions in which $\boldsymbol{\pi}$ can change in an optimization process, but the plane only has three DOFs. An optimizer is free to move the variable off the unit quaternion sphere.
- We use the exponential map from $\mathbb{R}^3$ to $\mathbb{S}^3$: [4] [5]

$$\exp(\boldsymbol{\zeta}) = \begin{cases} \left[\frac{1}{\|\boldsymbol{\zeta}\|}\sin(\frac{1}{2}\|\boldsymbol{\zeta}\|)\hat{\boldsymbol{\zeta}}, \sin(\frac{1}{2}\|\boldsymbol{\zeta}\|)\right]^T & \textit{if} \quad \boldsymbol{\zeta} \neq 0 \\ [0,0,0,1]^T & \textit{if} \quad \boldsymbol{\zeta} = 0 \end{cases} \tag{46}$$

with $\boldsymbol{\zeta} \in \mathbb{R}^3$ and $\hat{\boldsymbol{\zeta}} = \boldsymbol{\zeta}/\|\boldsymbol{\zeta}\|$.

- A plane $\boldsymbol{\pi} \in \mathbb{S}^3$ is updated by an increment $\boldsymbol{\zeta} \in \mathbb{R}^3$ using the quaternion multiplication

$$\boldsymbol{\pi}' = \exp(\boldsymbol{\zeta}) \circ \boldsymbol{\pi} \tag{47}$$

[4] M. Kaess, “Simultaneous localization and mapping with infinite planes”, 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4605-4611.

[5] Grassia, and F. Sebastian . “Practical Parameterization of Rotations Using the Exponential Map”. Journal of Graphics Tools 3.3(1998): 29-48.

Observation Model

- Plane observation model

$$\varphi_\pi(\boldsymbol{R}_{cg}, \boldsymbol{t}_{cg}, \boldsymbol{\pi}) = \begin{bmatrix} \boldsymbol{R}_{cg} & 0 \\ -\boldsymbol{t}_{cg}^T \boldsymbol{R}_{cg} & 1 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} \tag{48}$$

  with

$$\begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \frac{\boldsymbol{\pi}}{\pi_1^2 + \pi_2^2 + \pi_3^2} \tag{49}$$

  and $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3, \pi_4]^T$.
- ${}^c\boldsymbol{n}$ and ${}^c d$ are the unit normal of the plane and the vertical distance from the origin to the plane, respectively, expressed in the camera coordinate frame.
- $\boldsymbol{R}_{cg} \in \mathbb{SO}(3)$ and $\boldsymbol{t}_{cg} \in \mathbb{R}^3$ are the rotation matrix and the translation vector from the global coordinate frame to the camera coordinate frame, respectively.

Plücker Coordinates of 3D Lines

- The Plücker coordinates of a 3D line are denoted by $\boldsymbol{\mathcal{L}} = [\boldsymbol{u}^T, \boldsymbol{v}^T]^T \in \mathbb{P}^5$ and satisfy the Plücker constraint $\boldsymbol{u}^T \boldsymbol{v} = 0$.
- $\boldsymbol{u} \in \mathbb{R}^3$ is normal to the interpretation plane $\boldsymbol{\pi}_{\mathcal{L}}$ containing the line $\boldsymbol{\mathcal{L}}$.
- $\boldsymbol{v} \in \mathbb{R}^3$ is the line direction.
- The vertical distance from the origin to the line is computed by $\|\boldsymbol{u}\| / \|\boldsymbol{v}\|$.

Orthonormal Representation for 3D Lines

- The orthonormal representation [6] enables the local update of the Plücker coordinates using the minimum number of parameters.
- Any (projective) 3D line can be represented by [7]

$$(\boldsymbol{Q}, \boldsymbol{W}) \in \mathbb{SO}(3) \times \mathbb{SO}(2) \tag{50}$$

$(\boldsymbol{Q}, \boldsymbol{W})$ is the orthonomal representation of a 3D line.

[6] Bartoli, Adrien . "On the Non-Linear Optimization of Projective Motion Using Minimal Parameters". 7th European Conference on Computer Vision Springer-Verlag, 2002.

[7] Bartoli, Adrien , and P. Sturm . "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". Computer Vision and Image Understanding 100.3(2005):416-441.

Relating Plücker coordinates and orthonormal representation [8]

- The 3×2 matrix $[\boldsymbol{u}|\boldsymbol{v}]$ can be factored as

$$[\boldsymbol{u}|\boldsymbol{v}] = \boldsymbol{Q}\boldsymbol{\Sigma} \tag{51}$$

with

$$\begin{aligned} \boldsymbol{Q} &= \begin{bmatrix} \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|} & \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} & \frac{\boldsymbol{u}\times\boldsymbol{v}}{\|\boldsymbol{u}\times\boldsymbol{v}\|} \end{bmatrix} \in \mathbb{SO}(3) \\ \boldsymbol{\Sigma} &= \begin{bmatrix} \|\boldsymbol{u}\| & 0 \\ 0 & \|\boldsymbol{v}\| \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{52}$$

[8] Bartoli, Adrien , and P. Sturm . "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". Computer Vision and Image Understanding 100.3(2005):416-441.

Relating Plücker coordinates and orthonormal representation [9]

- Set

$$\boldsymbol{W} = \begin{bmatrix} \sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix} \in \mathbb{SO}(2) \tag{53}$$

with

$$\begin{aligned} \sigma_1 &= \frac{\|\boldsymbol{u}\|}{\sqrt{\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2}} \\ \sigma_2 &= \frac{\|\boldsymbol{v}\|}{\sqrt{\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2}} \end{aligned} \tag{54}$$

[9] Bartoli, Adrien , and P. Sturm . "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". Computer Vision and Image Understanding 100.3(2005):416-441.

Relating Plücker coordinates and orthonormal representation [10]

- The minimum four line parameters are denoted by $\boldsymbol{\eta} = [\boldsymbol{\theta}^T, \theta]^T$ where $\boldsymbol{\theta} \in \mathbb{R}^3$ and $\theta \in \mathbb{R}$.
- $\boldsymbol{Q}$ and $\boldsymbol{W}$ are updated by

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{Q} \exp([\boldsymbol{\theta}]_\times) \\ \boldsymbol{W} &= \boldsymbol{W} \exp([\theta]_\times) \end{aligned} \tag{55}$$

where $[\boldsymbol{\theta}]_\times$ and $[\theta]_\times$ are the 3×3 and 2×2 skew symmetric matrices corresponding to $\boldsymbol{\theta}$ and θ, respectively.

[10] Bartoli, Adrien , and P. Sturm . "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". Computer Vision and Image Understanding 100.3(2005):416-441.

Relating Plücker coordinates and orthonormal representation [11]

- Converting from the orthnormal representation $(\boldsymbol{Q}, \boldsymbol{W})$ to the Plücker coordinates $\boldsymbol{\mathcal{L}}$ by

$$\boldsymbol{\mathcal{L}} = [\sigma_1 \boldsymbol{q}_1^T, \sigma_2 \boldsymbol{q}_2^T]^T \tag{56}$$

where $\boldsymbol{q}_i$ is the i-th column of $\boldsymbol{Q}$.

[11] Bartoli, Adrien , and P. Sturm . "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment". Computer Vision and Image Understanding 100.3(2005):416-441.

Observation Model

- Line observation model

$$\varphi_{\mathcal{L}}\left(\boldsymbol{R}_{cg}, \boldsymbol{t}_{cg}, \boldsymbol{\mathcal{L}}\right) = \begin{bmatrix} \boldsymbol{R}_{cg} & [\boldsymbol{t}_{cg}]_{\times}\boldsymbol{R}_{cg} \\ 0 & \boldsymbol{R}_{cg} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{v}} \end{bmatrix} \tag{57}$$

where

$$\begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{v}} \end{bmatrix} = \frac{\boldsymbol{\mathcal{L}}}{\|\boldsymbol{v}\|} \tag{58}$$

- $\boldsymbol{R}_{cg} \in \mathbb{SO}(3)$ and $\boldsymbol{t}_{cg} \in \mathbb{R}^3$ are the rotation matrix and the translation vector from the global coordinate frame to the camera coordinate frame, respectively.
- $[\boldsymbol{t}_{cg}]_{\times}$ is the skew symmetric matrix corresponding to $\boldsymbol{t}_{cg}$.

Plane Fitting

- Minimizing the sum of squared distance from observed data points $\boldsymbol{p}_{\pi i}$ to the fitted plane model $(\boldsymbol{n}, d)$

$$E(\boldsymbol{n}, d) = \frac{1}{2} \sum_{i=1}^{N_\pi} \left(\boldsymbol{n}^T \boldsymbol{p}_{\pi i} + d \right)^2, \tag{59}$$

subject to $\boldsymbol{n}^T \boldsymbol{n} = 1$.

- By taking the partial derivative of $E(\boldsymbol{n}, d)$ w.r.t. d and setting it to zero, the optimal estimate of d can be computed by

$$d^* = -\boldsymbol{n}^T \boldsymbol{p}_{G\pi}. \tag{60}$$

with $\boldsymbol{p}_{G\pi}$ representing the centroid of all the observed points.

$$\boldsymbol{p}_{G\pi} = \frac{1}{N_\pi} \sum_{i=1}^{N_\pi} \boldsymbol{p}_{\pi i}. \tag{61}$$

Plane Fitting

- Substituting (60) into (59) yields

$$E(\boldsymbol{n}) = \frac{1}{2}\boldsymbol{n}^T \left(\sum_{i=1}^{N_\pi} (\boldsymbol{p}_{\pi i} - \boldsymbol{p}_{G\pi})(\boldsymbol{p}_{\pi i} - \boldsymbol{p}_{G\pi})^T \right) \boldsymbol{n}. \tag{62}$$

- Denote $\boldsymbol{S}_\pi = \sum_{i=1}^{N_\pi} (\boldsymbol{p}_{\pi i} - \boldsymbol{p}_{G\pi})(\boldsymbol{p}_{\pi i} - \boldsymbol{p}_{G\pi})^T$. The optimal estimate $\boldsymbol{n}^*$ equals the eigenvector of $\boldsymbol{S}_\pi$ corresponding to the smallest eigenvalue.
- The pseudoinverse of the covariance of the plane parameters $(\boldsymbol{n}, d)$ is estimated by the Hessian matrix [12] [13]

$$\boldsymbol{C}_\pi^\dagger = \boldsymbol{H}_\pi|_{\boldsymbol{n}^*, d^*} = \sum_{i=1}^{N_\pi} \begin{bmatrix} \boldsymbol{p}_{\pi i}\boldsymbol{p}_{\pi i}^T & \boldsymbol{p}_{\pi i} \\ \boldsymbol{p}_{\pi i}^T & 1 \end{bmatrix} \tag{63}$$

[12] Sivia, D.S.: Data Analysis: A Bayesian Tutorial. Oxford University Press (1996). DOI: ISBN0198518897.

[13] Pathak, K., Vaskevicius, N., Birk, A.: Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds. In: International Conference on Robotics and Automation (ICRA), pp. 1631 1636. IEEE press, Kobe, Japan (2009). DOI 10.1109/ROBOT.2009.5152502

Line Fitting

- Minimizing the sum of squared distance from observed data points $\boldsymbol{p}_{\mathcal{L}i}$ to the fitted line model $(\boldsymbol{u},\boldsymbol{v})$

$$E(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{2}\sum_{i=1}^{N_{\mathcal{L}}} \|\boldsymbol{u} - [\boldsymbol{p}_{\mathcal{L}i}]_{\times}\boldsymbol{v}\|^2, \tag{64}$$

subject to $\boldsymbol{v}^T\boldsymbol{v} = 1$ and $\boldsymbol{u}^T\boldsymbol{v} = 0$.

- By taking the partial derivative of $E(\boldsymbol{u},\boldsymbol{v})$ w.r.t. $\boldsymbol{u}$ and setting it to zero, the optimal estimate of $\boldsymbol{u}$ can be computed by

$$\boldsymbol{u}^* = [\boldsymbol{p}_{G\mathcal{L}}]_{\times}\boldsymbol{v}. \tag{65}$$

with

$$[\boldsymbol{p}_{G\mathcal{L}}]_{\times} = \frac{1}{N_{\mathcal{L}}}\sum_{i=1}^{N_{\mathcal{L}}}[\boldsymbol{p}_{\mathcal{L}i}]_{\times}. \tag{66}$$

Line Fitting

- Substituting (65) into (64) yields

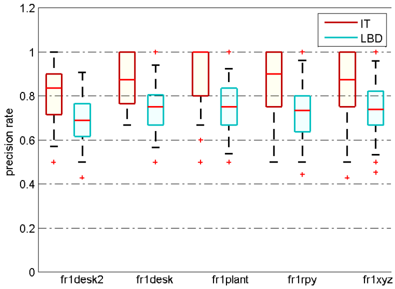
$$E(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^T \left(\sum_{i=1}^{N_{\mathcal{L}}} [\boldsymbol{p}_{\mathcal{L}i} - \boldsymbol{p}_{G\mathcal{L}}]_\times^T [\boldsymbol{p}_{\mathcal{L}i} - \boldsymbol{p}_{G\mathcal{L}}]_\times \right) \boldsymbol{v}. \tag{67}$$

- Denote $\boldsymbol{S}_{\mathcal{L}} = \sum_{i=1}^{N_{\mathcal{L}}} [\boldsymbol{p}_{\mathcal{L}i} - \boldsymbol{p}_{G\mathcal{L}}]_\times^T [\boldsymbol{p}_{\mathcal{L}i} - \boldsymbol{p}_{G\mathcal{L}}]_\times$. The optimal estimate $\boldsymbol{v}^*$ equals the eigenvector of $\boldsymbol{S}_{\mathcal{L}}$ corresponding to the smallest eigenvalue.
- The pseudoinverse of the covariance of the line parameters $(\boldsymbol{u}, \boldsymbol{v})$ is estimated by the Hessian matrix

$$\boldsymbol{C}_{\mathcal{L}}^\dagger = \boldsymbol{H}_{\mathcal{L}}|_{\boldsymbol{u}^*, \boldsymbol{v}^*} = \sum_{i=1}^{N_{\mathcal{L}}} \begin{bmatrix} \boldsymbol{I}_3 & [\boldsymbol{p}_{\mathcal{L}i}]_\times^T \\ [\boldsymbol{p}_{\mathcal{L}i}]_\times & [\boldsymbol{p}_{\mathcal{L}i}]_\times^T [\boldsymbol{p}_{\mathcal{L}i}]_\times \end{bmatrix} \tag{68}$$

Experiments

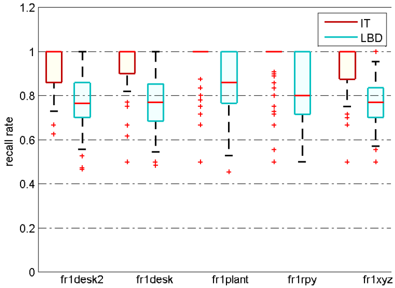
- Experiment 1: Line Matching (LM)
  - methods: IT-LM and LBD-LM [14]
  - precision rate of each frame-to-frame association



[14] R. Gomez-Ojeda, F. Moreno, D. Zuiga-Nol, D. Scaramuzza and J. Gonzalez-Jimenez, ”PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments,” in IEEE Transactions on Robotics, vol. 35, no. 3, pp. 734-746, June 2019.

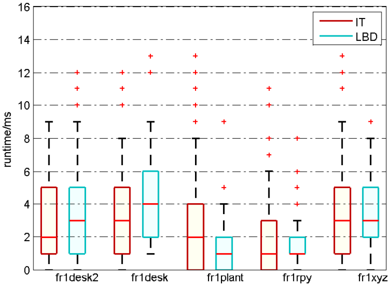
Experiments

- Experiment 1: Line Matching (LM)
  - methods: IT-LM and LBD-LM
  - recall rate of each frame-to-frame association



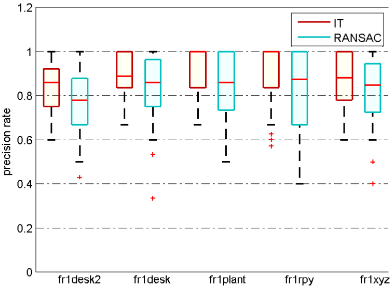
Experiments

- Experiment 1: Line Matching (LM)
  - methods: IT-LM and LBD-LM
  - runtime of each frame-to-frame association



Experiments

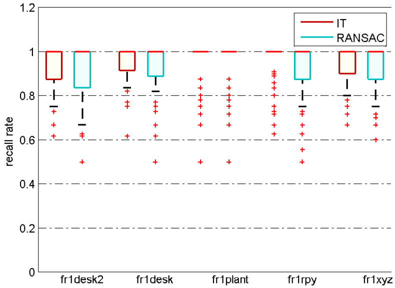
- Experiment 2: Geometric Feature (Plane and Line) Matching (GFM)
  - methods: IT-GFM and RANSAC-GFM [15]
  - precision rate of each frame-to-frame association



[15] C. Raposo, M. Antunes and J. P. Barreto, "Piecewise-Planar StereoScan: Sequential Structure and Motion Using Plane Primitives," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 8, pp. 1918-1931, 1 Aug. 2018.

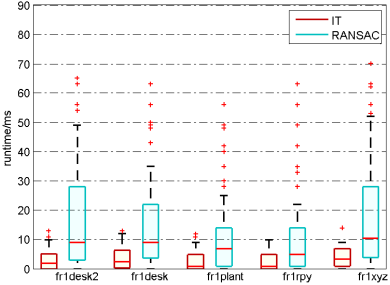
Experiments

- Experiment 2: Geometric Feature (Plane and Line) Matching (GFM)
  - methods: IT-GFM and RANSAC-GFM
  - recall rate of each frame-to-frame association



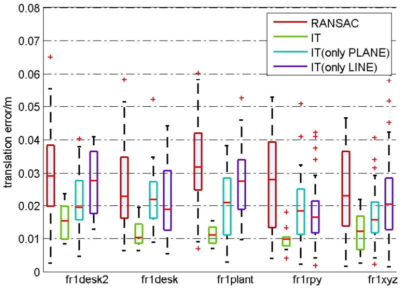
Experiments

- Experiment 2: Geometric Feature (Plane and Line) Matching (GFM)
  - methods: IT-GFM and RANSAC-GFM
  - runtime of each frame-to-frame association



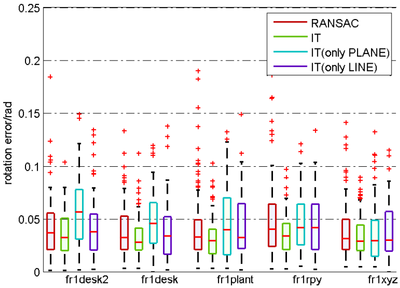
Experiments

- Experiment 3: Geometric Feature (Plane and Line) Matching (GFM)
  - methods: IT-GFM, IT-Plane-GFM, IT-Line-GFM and RANSAC-GFM
  - translation error



Experiments

- Experiment 3: Geometric Feature (Plane and Line) Matching (GFM)
  - methods: IT-GFM, IT-Plane-GFM, IT-Line-GFM and RANSAC-GFM
  - rotation error



Experiments

- Experiment : VO
  - methods: IT-GFM-VO, Prob-RGBD-VO [16]
  - performance: ATE, RPE
  - benchmarks: the TUM benchmark and ICL-NUIM benchmark

Table 1: Comparison of VO.

	ATE		RPE	
	IT-GFM-VO	Prob-RGBD-VO	IT-GFM-VO	Prob-RGBD-VO
fr1/desk	0.047m	0.040m	0.062m;3.7deg	0.023m;1.7deg
fr1/360	**0.088m**	0.091m	**0.062m;2.1deg**	0.064m;2.7deg
fr3/str_notext_far	**0.038m**	0.054m	**0.014m;0.5deg**	0.019m;0.7deg
fr3/cabinet	**0.051m**	0.200m	**0.016m;1.3deg**	0.039m;1.8deg
lr0	**0.058m**	0.059m	0.011m;0.3deg	0.006m;0.5deg
or0	**0.085m**	0.106m	0.007m;0.5deg	0.006m;0.5deg

[16] Pedro F. Proena, Yang Gao, Probabilistic RGB-D odometry based on points, lines and planes under depth uncertainty, Robotics and Autonomous Systems, Volume 104, 2018, Pages 25-39.

Experiments

- Experiment : SLAM
  - methods: IT-GFM-SLAM, CPA-SLAM [17] , GC-SLAM [18]
  - performance: ATE
  - benchmarks: the TUM benchmark and ICL-NUIM benchmark.

Table 2: Comparison of SLAM.

	ATE		
	IT-GFM-VO	CPA-SLAM	GC-SLAM
fr1/desk	0.026m	0.018m	0.019m
fr2/xyz	**0.011m**	0.014m	0.011m
fr3/office	**0.022m**	0.025m	0.026m
fr3/nst	**0.016m**	0.016m	0.016m
lr2noisy	0.013m	0.089m	0.008m
lr3noisy	0.016m	0.009m	0.010m

[17] L. Ma, C. Kerl, J. Stückler and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 2016, pp. 1285-1291.

[18] L. Han, L. Xu, D. Bobkov, E. Steinbach and L. Fang, "Real-Time Global Registration for Globally Consistent RGB-D SLAM," in IEEE Transactions on Robotics, vol. 35, no. 2, pp. 498-508, April 2019.