

## 第 3 章

# 複数 LiDAR の位置校正方法とオクルージョン範囲の導出

### 3.1 はじめに

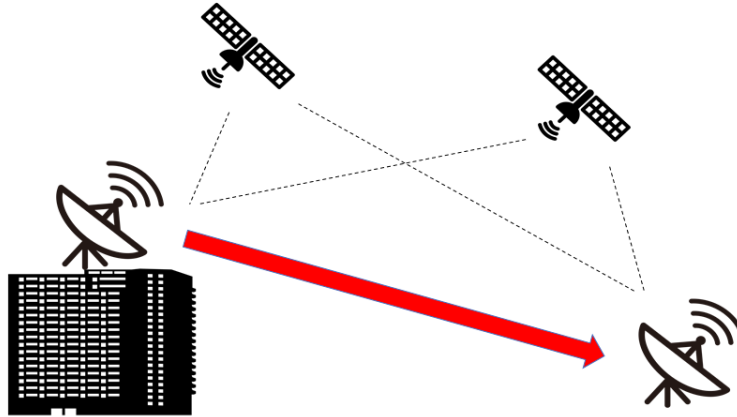
本章では，LiDAR の位置を取得するために使用した手法を説明する．また，決定した LiDAR の位置とそれが認識した人物の位置からオクルージョンが起きている範囲を導出し，その範囲を UAV が撮影できるように目標位置を設定する手法を説明する．

### 3.2 複数 LiDAR の位置決定方法

まず始めに，複数 LiDAR の位置の取得方法について述べる．先行研究では 1 台の LiDAR の位置を UAV に搭載している GPS センサと同じものを使用して取得していた．しかし，UAV に搭載されている GPS センサから得られる緯度経度情報は数 m から十数 m の誤差がある．複数の LiDAR を使用する際，位置の誤差が大きいと，1 つの物体を 2 つの物体と誤認してしまう恐れがある．そのため，2 つの LiDAR の正確な相対位置を取得するために RTK-GPS 測位という手法を利用して位置を取得する．

#### 3.2.1 RTK-GPS 測位

RTK-GPS(Real-Time Kinematic GPS) 測位とは，位置が分かっている移動しない基地局 (Base) と位置情報を取得しようとしている観測点である移動局 (Rover) で同時に GPS 観測を行い，基準局で観測したデータを移動局へリアルタイムに送信し，基準局の位置に基づい



**Fig. 3.1 Image of RTK-GPS Positioning System**

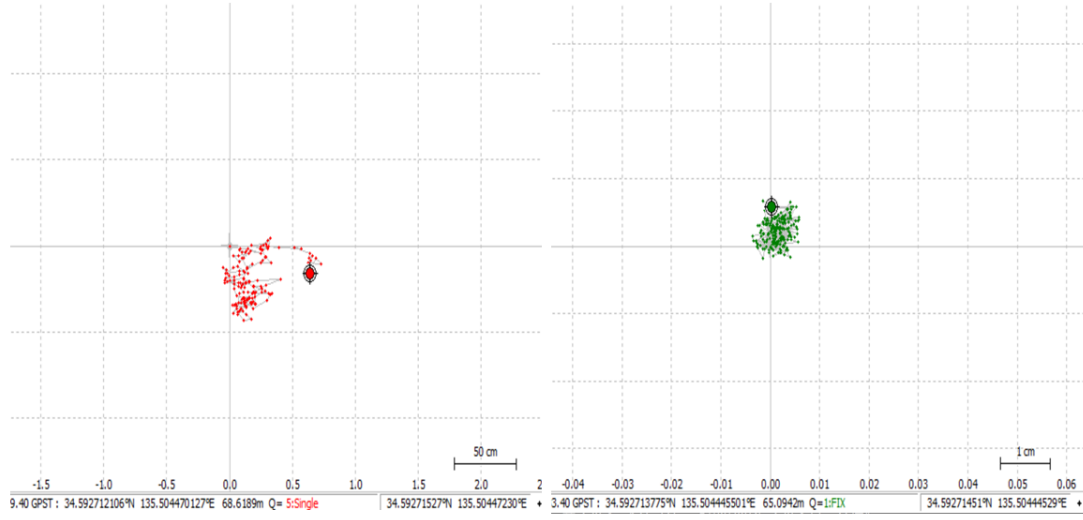
て移動局の位置を求める手法である．さらにネットワークを利用して基地局と移動局のデータ送信を行うことで，基地局と移動局が長距離で離れていても精度の高い演算ができる．

### 3.2.2 RTK-GPS 測位と単独測位の比較

単独測位では数 m から十数 m の誤差が発生するのに対して，RTK-GPS 測位では数 cm の誤差が発生するといわれている．ここで，実際に計測したデータを比較して測位の性能の差を述べる．計測位置は大阪市立大学 F 棟 507 号室のベランダであり，Fig. 3.2 は 30 秒の計測データをグラフ化したものである．Fig. 3.2 左のグラフは単独測位の結果であり，グラフは 1 マス 50cm である．同図右のグラフは RTK-GPS 測位の結果であり，グラフは 1 マス 1cm である．グラフはから見て取れるように，単独測位は 50cm から 1m の誤差があり，RTK-GPS 測位の誤差は 1cm から 2cm 以内に収まっている．

## 3.3 GPS 座標から map 座標への変換

本研究では，東を  $x$  座標正方向，北を  $y$  座標正方向とする map 座標系を用いる．また座標原点は LiDAR1 の座標をもとに算出する．ここでは，GPS センサから得た LiDAR1 の緯度経度と座標から原点の緯度経度の値の算出方法，原点の緯度経度の値と LiDAR2 の緯度



**Fig. 3.2 Single GPS Positioning (right) and RTK-GPS Positioning (left)**

経度の値から LiDAR2 の map 座標の算出方法を述べる.

LiDAR1 の GPS 座標を  $lon_{lidar1}, lat_{lidar1}$ , map 座標を  $x_{lidar1}, y_{lidar1}$  とすると求める map 座標原点の緯度経度の値  $lon_{origin}, lat_{origin}$  は以下の式 (3.1) で算出される. なお式中の  $R$  は地球の赤道半径である.

$$\begin{aligned} lat_{origin} &= \frac{y_{lidar1}}{R} \times \frac{180}{\pi} + lat_{lidar1} \\ lon_{origin} &= \frac{x_{lidar1}}{R} \times \frac{180}{\pi} \times \frac{1}{\cos(lat_{origin} \frac{180}{\pi})} + lon_{origin} \end{aligned} \quad (3.1)$$

式 (3.1) より得られた map 座標原点の緯度経度の値を用いて, LiDAR2 の map 座標  $x_{lidar2}, y_{lidar2}$  を式 (3.2) で求めることができる. LiDAR2 の GPS 座標を  $lon_{lidar2}, lat_{lidar2}$  とする.

$$\begin{aligned} x_{lidar2} &= R(lon_{lidar2} - lon_{origin}) \frac{\pi}{180} \cos((lat_{lidar2} - lat_{origin}) \frac{\pi}{180}) \\ y_{lidar2} &= R(lat_{lidar2} - lat_{origin}) \frac{\pi}{180} \end{aligned} \quad (3.2)$$

### 3.4 オクルージョンが発生した場合の UAV の撮影位置の決定

前節まででは、LiDAR の位置を取得するための手法を述べた。人物行動範囲内に複数人の人物が存在する場合、オクルージョンが発生し、地上設置 LiDAR では人物行動範囲内の全ての人物をとらえきれない場合が存在する。この節では、オクルージョンが発生した場合の LiDAR の撮影できない領域と、その領域を考慮した UAV の目標位置の導出方法について述べる。

#### 3.4.1 数値計算安定化のための各種パラメータ設定方法

問題で用いる係数行列およびベクトルを次のように設定することで、反復計算の安定性および収束性能を向上させる。

- $K_p$ ,  $K_u$ ,  $K_{u_z}$  の各成分を、係るベクトルの各要素をそれぞれ最大値で正規化するように設定する

$$\begin{aligned}
 K_p &= \text{diag} \{k_x, k_y, k_\theta, k_l, k_{\theta_{fl}}, k_{\theta_{fr}}\} \\
 k_x &= k_y = (\max\{|x_{\text{goal}} - x^0|, |y_{\text{goal}} - y^0|\})^{-2} \\
 k_\theta &= (2\pi)^{-2} \\
 k_l &= (Y_{\max} - Y_{\min})^{-2} \\
 k_{\theta_{fl}} &= k_{\theta_{fr}} = (2\pi)^{-2}
 \end{aligned} \tag{3.3}$$

$$\begin{aligned}
 K_u &= \text{diag} \{\dots, k_{u_A}, k_{u_B}, k_{u_{l1}}, k_{u_{l2}}, k_{u_{fl}}, k_{u_{fr}}, \dots\} \\
 k_{u_A} &= k_{u_B} = (2 \tan^{-1}(X_{\max}/Y_{\min}) + \theta_{2\max})^{-2} \\
 k_{u_{l1}} &= k_{u_{l2}} = (Y_{\max} - Y_{\min})^{-2} \\
 k_{u_{fl}} &= k_{u_{fr}} = (\theta_{2\max} - \theta_{2\min})^{-2}
 \end{aligned} \tag{3.4}$$

$$\begin{aligned}
 K_{u_z} &= \text{diag} \{\dots, k_{u_{zl}}, k_{u_{zr}}, \dots\} \\
 k_{u_{zl}} &= k_{u_{zr}} = z_{\max}^{-2}
 \end{aligned} \tag{3.5}$$

- $K_u$ ,  $K_{u_z}$  は、前項の設定に加えて Sugihara の手法 [?] を適用し、評価関数??の値を用いて重み付けする ( $I$  は単位行列,  $\Delta k_u$  はスカラー定数)

$$\begin{aligned}
K_u &\leftarrow V_{\text{now}} K_u + \Delta k_u I = \frac{1}{2} \|\epsilon_{\text{now}}\|_{K_p}^2 K_u + \Delta k_u I \\
K_{u_z} &\leftarrow V_{\text{now}} K_{u_z} + \Delta k_{u_z} I = \frac{1}{2} \|\epsilon_{\text{now}}\|_{K_p}^2 K_{u_z} + \Delta k_{u_z} I
\end{aligned} \tag{3.6}$$

- 一つ一つの制約条件  $\omega_i(\mathbf{q}^0, \mathbf{U}_k) \geq 0$  を  $\|\nabla \omega_i\|$  で正規化することにより、GI 法内部での反復計算を安定化させ、また緩和問題での各制約条件の比重を揃える [?]
- 緩和問題を解く際、評価関数全体を  $\|\phi\|$  で正規化し、元の評価関数部分の値とスラック変数の値の比重を揃える ( $\rho = 1$  とできる)

そして、得られた解を用いて Armijo の基準に基づく直線探索を行い、入力列を更新する。

### 3.4.2 直線探索

直線探索では、次式を満たすステップ幅  $\alpha$  を求める。

$$\begin{aligned}
H(\mathbf{q}^0, \mathbf{U}_k + \alpha \Delta \mathbf{U}_k) &\leq H(\mathbf{q}^0, \mathbf{U}_k) + \beta \alpha \nabla V(\mathbf{q}^0, \mathbf{U}_k)^T \Delta \mathbf{U}_k \\
&= H(\mathbf{q}^0, \mathbf{U}_k) + \beta \alpha \phi^T \Delta \mathbf{U}_k
\end{aligned} \tag{3.7}$$

$\beta$  は、 $0 < \beta < 1$  を満たす定数である。今回、 $\alpha$  は??に示す単純な探索により求める。なお、 $\gamma$  は  $0 < \gamma < 1$  を満たす減衰係数である。

## 3.5 終了判定

更新後、終了判定を行う。次の項目をすべて満たすことを終了条件とする。

- $\epsilon$  が収束（各要素が許容誤差  $\epsilon_{\text{goal}}$  の各要素よりも小さい）
- $\Delta \mathbf{U}_k$  が収束（各要素が許容誤差  $\epsilon_{\text{input}}$  の各要素よりも小さい）
- $\omega_i(\mathbf{q}^0, \mathbf{U}_k) \geq 0$  ( $i = 0, \dots, 20k - 1$ ) をすべて満たす

これらの条件を全て満たしていれば、その時点での  $\mathbf{U}_k$  を解とし、計画を終了する。満たしていない場合は次節の処理へ移る。

## 3.6 目標到達可否判定

初期ステップ数  $k_{\text{init}}$  は直線距離から算出するため、現在の  $k$  では歩数不足の場合がある。そこで、反復毎に目標状態へ収束する見込みがあるかどうかを判定し、見込みがない場合に  $k$  を増やす処理を行う。ここでは、次の項目のいずれかを満たせば到達不可と判定すること

とする.

- “ $\Delta U_k$  収束時に  $\epsilon$  が未収束” という状況が  $n_{\text{reachable1}}$  回繰り返される
- 現在の  $k$  における反復回数  $\text{loop}$  が  $n_{\text{reachable2}}$  に達した段階で,  $\epsilon$  が大きい値をとる (各要素が到達見込みの閾値  $\epsilon_{\text{reachable}}$  の各要素よりも大きい)

到達不可と判定された場合には  $k \leftarrow k + 1$  とし, また  $U_k$  はその内容を引き継ぎ  $U_k \leftarrow (U_k^T, \mathbf{0}^T)^T$  として再度反復を開始する. そうでなければ, “Solve QPP” へ戻り反復を継続する.

### 3.7 地形情報の構築

前述の計画アルゴリズムを開始する前に, 地形適応制約で用いる地形情報  $z_{\text{field}}(x, y)$  を構築する必要がある. 深度カメラや測域センサから得られる点群データを用いて地形情報を構築することを想定し, 次の手順で点群データからの構築を行った.

1.  $xy$  平面を等間隔に幅  $D_{\text{grid}}$  四方の正方形グリッドで分割する.
2. 各グリッドの領域に含まれるサンプルデータ群の  $z$  軸成分の平均値を, グリッド中央における高さとする.
3.  $x, y$  方向それぞれについて, グリッド中心の点列をサンプルとする 3 次スプライン補間関数を計算.
4.  $z_{\text{field}}(x, y), \partial z_{\text{field}}/\partial x, \partial z_{\text{field}}/\partial y$  は, 与えられた点  $(x, y)$  に対して直近の  $x$  軸方向 2 本,  $y$  軸方向 2 本の補間関数上の値の距離重み付け平均で算出する.

なお, シミュレーションにおいて関数からサンプリングを行って点群データを得る場合,  $D_{\text{sample}}$  間隔でサンプリングを行う.

### 3.8 本章のまとめ

本章では, 二次計画法を用いた反復法による脚配置計画問題の求解アルゴリズムの要素として, 必要歩数の推定から毎回の二次計画問題の求解と緩和問題の設定, 収束判定, 収束見込み判定とステップ数追加について説明した. また, 地形の違いなどにより問題ごとに大きく変化してしまうことを可能な限り回避するため, 反復法における数値計算の性能を安定させる重み行列の設定方法などについても述べた.

次章以降は, 提案する脚配置計画手法の有用性および性能を検証するための計画シミュレーションおよび実機実験について述べていく.