第 2 章

地上固定 LiDAR と複数 UAV を用いた撮影計画の概要

2.1 はじめに

本章では，昨年度まで佐々木らが行っていた地上固定 LiDAR と複数 UAV を用いた撮影計画の概要について述べる．[12] またその研究における問題点，課題点などについて述べる．

2.2 撮影ベクトルと移動カメラの撮影領域

2.2.1 撮影ベクトル

カメラで撮影対象を撮影するとき，撮影対象の表面に対して垂直な方向から撮影することが望ましい．撮影対象の表面に垂直な方向を撮影ベクトルと定義する．また定義した撮影ベクトルに対して，一定の解像度以上で撮影できる領域を撮影領域と定義する．要求解像度と UAV に搭載されている移動カメラの性能から撮影領域は計算できる．

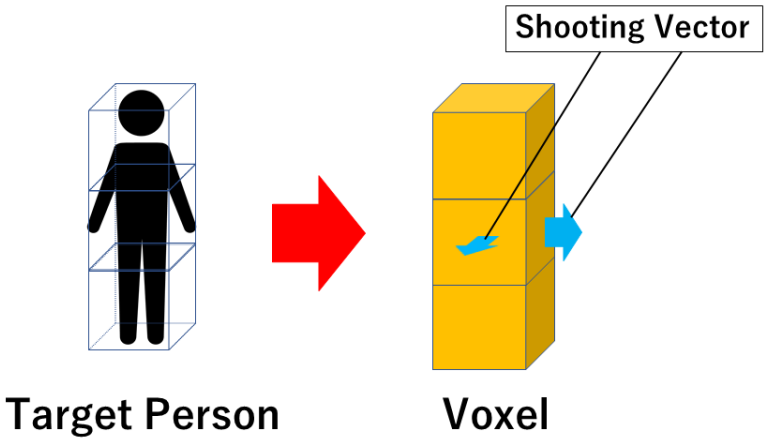

Fig. 2.1　Model of Shooting Vector

撮影領域の導出において，三次元空間および二次元空間はグリッドによって分割され，それぞれボクセル，セルとして管理される．Fig. 2.1 は撮影対象人物とその人物に対するボクセルと撮影ベクトルである．この研究では Fig. 2.1 で中心のセルから撮影ベクトルが表されているように，簡単のため，対象人物をは 1 つのボクセル内で存在するものとし，その 1 つのボクセルから垂直な方向に撮影ベクトルを定義する．

2.2.2　要求解像度

この研究では，1[mm] を表すために必要な画素数 [pixel/mm] で解像度を表現する．あるカメラが撮影対象を撮影しているとき，カメラの水平 (垂直) 解像度を R_{camera}，カメラの水平 (垂直) 画角を α[rad]，撮影対象を写している水平 (垂直) 画角を β とし，セルの 1 辺の長さが S[mm] であるときの，撮影対象の解像度 R[pixel/mm] は式 (2.1) で求められる．

$$\frac{\beta R_{camera}}{\alpha S}\,[\text{pixel/mm}] \tag{2.1}$$

任意の要求解像度 n[pixel/mm] が与えられたとき，要求解像度の満たすための条件は式 (2.1) より $n \leq R$ で表される．この式より，要求解像度を満たすためには画角 β に対して式 (2.2) を満足する必要がある．

$$\beta \leq \frac{\alpha n S}{R_{camera}} [\text{rad}] \tag{2.2}$$

以上より，ある撮影ベクトルについて要求解像度を満たす撮影領域は，条件式 (2.2) を満たす水平画角 β_h，垂直画角 β_v の点の軌跡である．

2.2.3 撮影領域の導出

撮影領域の導出を平面について順番に導出していく．まず始めに x，y 平面について考える．撮影ベクトルが設定されたセルの中心を原点とする x，y 座標系を考える．カメラの座標を (x,y)[mm]，撮影ベクトルと x 軸のなす角を γ，セルの 1 辺の長さを 2λ[mm] とすると水平画角 β_h はのようになる．

$$\beta_h = \left| \arctan\left(\frac{y + \lambda\cos\gamma}{x - \lambda\sin\gamma}\right) - \arctan\left(\frac{y - \lambda\cos\gamma}{x + \lambda\sin\gamma}\right) \right| \tag{2.3}$$

簡単のために $\gamma = 0$ のときを考えると，xy 平面における水平画角 β_h で捉えることのできる点の軌跡は式 (2.4) のようになる．

$$\left(x - \frac{\lambda}{\tan\beta_h}\right)^2 + y^2 = \left(\frac{\lambda}{\tan\beta_h}\right)^2 + \lambda^2 \tag{2.4}$$

2.3　脚先の高さの取り扱い

2.4　計画問題の定式化と解法

??より，k ステップ後の状態 $\boldsymbol{q}^k$ は初期状態 $\boldsymbol{q}^0$ と入力列 $\boldsymbol{U}_k = (\boldsymbol{u}^{0^T}, \ldots, \boldsymbol{u}^{k-1^T})^T$ を用いて $\boldsymbol{q}^k = \boldsymbol{G}_k(\boldsymbol{q}^0, \boldsymbol{U}_k)$ と表すことができる．目標状態 $\boldsymbol{q}_{\text{goal}}$ へ k ステップで到達するとしたときの脚配置計画問題は，式 (2.5) の非線形代数方程式の解 $\boldsymbol{U}_k$ を求める逆運動学問題となる．

$$\boldsymbol{q}_{\text{goal}} = \boldsymbol{G}_k\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) \tag{2.5}$$

この問題式 (2.5) を $\boldsymbol{U}_k$ について解けば脚配置計画が求まる．

本論文では，目標追従を評価関数で表し，制約条件下での最適化問題として解くことを考える．$\boldsymbol{U}_k$，$\boldsymbol{q}^0$ を変数とする評価関数を $V = H(\boldsymbol{q}^0, \boldsymbol{U}_k)$，制約条件を $\boldsymbol{\omega}(\boldsymbol{q}^0, \boldsymbol{U}_k) \geq \boldsymbol{0}$ として，式 (2.6) の最適化問題を定義する．

$$\begin{aligned} &\text{minimize} \quad V = H\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) \\ &\text{subject to} \quad \boldsymbol{\omega}\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) \geq \boldsymbol{0} \end{aligned} \tag{2.6}$$

この式 (2.6) の非線形計画問題を，反復法を用いて解くことを考える．反復のある段階での入力列 $\boldsymbol{U}_k$ に微小変化 $\boldsymbol{\Delta U}_k$ を加えて $\boldsymbol{U}_k \leftarrow \boldsymbol{U}_k + \boldsymbol{\Delta U}_k$ として更新するとき，更新前の評価値 V_{now} は，更新後の評価値 V_{next} と変化量 ΔV を用いて $V_{\text{next}} = V_{\text{now}} + \Delta V$ と表すことができる．線形近似を用いて V_{next} を $\boldsymbol{\Delta U}_k$ に関する二次形式，制約条件を一次式に近似し，更新量 $\boldsymbol{\Delta U}_k$ を求める問題を式 (2.7) の制約条件付きの二次計画問題に定式化する．

$$\begin{aligned} &\text{minimize} \quad \Delta V = \frac{1}{2}\,\boldsymbol{\Delta U}_k^T\,\boldsymbol{\Phi}\,\boldsymbol{\Delta U}_k + \boldsymbol{\phi}^T\,\boldsymbol{\Delta U}_k \\ &\text{subject to} \quad \boldsymbol{\Omega}\,\boldsymbol{\Delta U}_k + \boldsymbol{\omega} \geq \boldsymbol{0} \end{aligned} \tag{2.7}$$

この問題式 (2.7) を逐次解いて $\boldsymbol{U}_k$ を更新していくことで，計画問題の解が得られる．

なお, 以降では入力列更新前と後の変数をそれぞれ $(\text{variable\_name})_{\text{now}}$, $(\text{variable\_name})_{\text{next}}$ の形で表すこととする．

2.5 評価関数

本論文では，目標状態到達を計画の指標とし，最適化する評価関数には目標追従二乗誤差を用いる．

$$V = \frac{1}{2}\|\varepsilon\|^2 = \frac{1}{2}\left\|\boldsymbol{q}_{\text{goal}} - \boldsymbol{q}^k\right\|_{K_p}^2 \tag{2.8}$$

K_p は重み行列である．これを式 (2.6) の問題の評価関数として設定する．なお，目標追従以外の指標（ [**?**, **?**] における運動エネルギなど）も評価関数に加える事で最適化可能である．

問題式 (2.7) へ適用するため，$\boldsymbol{\Delta U}_k$ によって更新された後の状態 $\boldsymbol{q}^k_{\text{next}}$ を，$\boldsymbol{q}^k_{\text{now}}$ の $\boldsymbol{U}_k$ についてのヤコビ行列 $J_k = \partial \boldsymbol{G}_k / \partial \boldsymbol{U_k}$ を用いて線形近似する．

$$\boldsymbol{q}^k_{\text{next}} = \boldsymbol{q}^k_{\text{now}} + J_k\, \boldsymbol{\Delta U}_k \tag{2.9}$$

式 (2.9) を用いて二次計画問題式 (2.7) の評価関数を式 (2.10) で定義する．

$$\begin{aligned}
V_{\text{next}} &= \frac{1}{2}\|\varepsilon_{\text{next}}\|_{K_p}^2 + \frac{1}{2}\|\boldsymbol{\Delta U}_k\|_{K_u}^2 + \frac{1}{2}\left\|\boldsymbol{\Delta U}_{zk}\right\|_{K_{uz}}^2 \\
&= \frac{1}{2}\left(\varepsilon_{\text{now}} - J_k\, \boldsymbol{\Delta U}_k\right)^T K_p \left(\varepsilon_{\text{now}} - J_k\, \boldsymbol{\Delta U}_k\right) \\
&\quad + \frac{1}{2}\, \boldsymbol{\Delta U}_k^T K_u\, \boldsymbol{\Delta U}_k \\
&\quad + \frac{1}{2}\left(J_{zk}\, \boldsymbol{\Delta U}_k\right)^T K_{u_z} \left(J_{zk}\, \boldsymbol{\Delta U}_k\right) \\
&= \frac{1}{2}\, \boldsymbol{\Delta U}_k^T \Phi\, \boldsymbol{\Delta U}_k + \phi^T\, \boldsymbol{\Delta U}_k + \text{const.} \\
\Phi &= J_k^T\, K_p J_k + K_u + J_{zk}^T\, K_{u_z} J_{zk} \\
\phi &= -J_k^T\, K_p\, \varepsilon_{\text{now}}
\end{aligned} \tag{2.10}$$

K_p，K_u，K_{u_z} は評価関数の各項を重み付ける対角行列である（対角成分の設定方法は**??**に示す）．$\phi \in R^{6k}$ はベクトル，$\Phi \in R^{6k \times 6k}$ は実対称の正定値行列である．

式 (2.10) の式変形前の V_{next} の定義において，第一項は前述の目標追従二乗誤差の評価項，第二項は入力列の更新量 $\Delta \boldsymbol{U}_k$ 最小化の評価項である．第三項は，高さ変位を最小化するための項である．脚配置モデルが状態として z_l^i，z_r^i を持つと仮定すると，式 (2.11) のように次ステップへ遷移すると考えられる．

$$
\begin{aligned}
z_l^{i+1} &= z_l^i + u_{zl}^i \\
z_r^{i+1} &= z_r^i + u_{zr}^i
\end{aligned}
\tag{2.11}
$$

高さに対する仮想的な入力 u_{zl}^i，u_{zr}^i は**??**を用いて $\boldsymbol{q}^i$，$\boldsymbol{q}^{i+1}$，$z_{\text{field}}(x, y)$ から計算でき，高さに対する仮想入力列 $\boldsymbol{U}_{zk} = (u_{zl}^0, u_{zr}^0, \ldots, u_{zl}^{k-1}, u_{zr}^{k-1})^T$ は $\boldsymbol{q}^0$，$\boldsymbol{U}_k$ で表すことができる．入力列更新時の $\boldsymbol{U}_{zk}$ の変化 $\Delta \boldsymbol{U}_{zk}$ を最小化すれば，高さ方向の変位を最小化する計画が得られる．なお，式 (2.10) 中の J_{zk} は，$\boldsymbol{U}_{zk}$ の $\boldsymbol{U}_k$ についてのヤコビ行列 $J_{zk} = \partial \boldsymbol{U}_{zk} / \partial \boldsymbol{U}_k$ である．

2.6 制約条件

制約条件は，三次元上の脚配置を実現するために必要となる地形との接触についての制約と，適用するロボットに合わせて定式化されるハードウェア上の制約からなる．

2.6.1 地形との接触制約

平面上で定義されたモデルを用いて三次元の脚配置を実現するため，地形との接触について制約を課す必要がある．脚の垂直方向の移動が平面動作から独立していること，機構的に踏み出し時の高さには限界があることを踏まえ，**??**を用いて両脚間の高低差について制約を

課す．

$$\begin{aligned} -z_{\max} \leq z_l^{i+1} - z_r^i \leq z_{\max} \\ -z_{\max} \leq z_r^{i+1} - z_l^{i+1} \leq z_{\max} \end{aligned} \tag{2.12}$$

$z_{\max}$ は高低差の絶対値上限である．この制約を課すと，歩行時の垂直方向の移動が上下限値内に収まるように歩幅を抑えた脚配置が求まることになり，地形を考慮した脚配置計画が実現される．地形形状に関しては，平面位置に対応する高さ $z_{\text{field}}(x, y)$，線形近似時に必要となる x，y 方向それぞれの傾き $\partial z_{\text{field}}/\partial x$, $\partial z_{\text{field}}/\partial y$ が既知であればよい．上限値 $z_{\max}$ は，定数としても関数としてもよい．即ち，踏み出しの高さによって遊脚の水平方向の可動範囲が変化するような場合には，$\boldsymbol{q}^i$，$\boldsymbol{u}^i$ および z_{field} を用いて表現される関数とするなど，適用するロボットに合わせて定式化を行えばよい．

なお，現在は地形の傾斜に対する脚先の姿勢を考慮していない．今後，脚裏形状や制御における許容範囲，地形形状情報から定式化できると考えている．

2.6.2 遊脚の配置制約

制約条件を定式化するために，**??**，**??**および**??**中に示している，次の三つのパラメータを定義する．

- 支持脚と遊脚の相対距離 L^i
- 支持脚の y 軸から見た直線（L^i）の傾き θ_1^i
- 支持脚を中心としたときの両脚先間の相対角度 θ_2^i

L^i，θ_1^i，θ_2^i はそれぞれ，右脚支持の場合に $L^i = l^i + u_{l1}^i$，$\theta_1^i = u_A^i - \theta^i + \theta_{fr}^i$，$\theta_2^i = \theta_{fl}^{i+1} - \theta_{fr}^i$，左脚支持の場合に $L^i = l^{i+1}$，$\theta_1^i = \theta^{i+1} - \theta_{fl}^{i+1}$，$\theta_2^i = \theta_{fl}^{i+1} - \theta_{fr}^{i+1}$ となる．

まず，支持脚に対する遊脚の相対角度 θ_2^i に制約を課す．これは式 (2.13) で定式化される（**??**参照）．

$$\theta_{2\min} \leq \theta_2^i \leq \theta_{2\max} \tag{2.13}$$

$\theta_{2\max}$，$\theta_{2\min}$ は相対角度の上下限値である．

次に，支持脚に対する遊脚の可動範囲を**??**に示す領域内に制限する制約を課す．支持脚に対する遊脚の相対位置座標 $(x^i_{sw}, y^i_{sw}) = (L^i \sin\theta^i_1, L^i \cos\theta^i_1)$ は，$x^i_{sw} \geq 0$，$x^i_{sw} < 0$ それぞれの範囲において，$(0, Y_{\min})$ を中心とする楕円形の領域に制限されることとなる．この楕円の径については，$x^i_{sw} \geq 0$，$x^i_{sw} < 0$ それぞれの領域において，長軸は共通だが短軸は異なる．

$$
\begin{gathered}
y^i_{sw} \geq Y_{\min} \\
\left(\frac{x^i_{sw}}{X}\right)^2 + \left(\frac{y^i_{sw} - Y_{\min}}{Y_{\max} - Y_{\min}}\right)^2 \leq 1.0 \\
\left(X = \begin{cases} X_{\max} & \left(x^i_{sw} \geq 0\right) \\ X_{\min} & \left(x^i_{sw} < 0\right) \end{cases} \right)
\end{gathered}
\tag{2.14}
$$

$Y_{\max}$，$Y_{\min}$ は Σ_{sp} の y 軸上の両脚間隔の上限値及び下限値．また，$X_{\max}$，$X_{\min}$ は，脚間距離が最小のときの前進・後進それぞれについての Σ_{sp} の x 軸方向の歩幅上限値．

2.6.3 脚先の干渉防止制約

さらに，脚先が重なることを防ぐための制約を課す．**??**に示している，支持脚座標系 Σ_{sp} から見た遊脚の頂点 $\boldsymbol{R}^i_A = (x^i_{R_A}, y^i_{R_A})$ と $\boldsymbol{R}^i_B = (x^i_{R_B}, y^i_{R_B})$，また遊脚座標系 Σ_{sw} から見た

左脚の頂点 $\boldsymbol{L}_A^i = (x_{L_A}^i, y_{L_A}^i)$ と $\boldsymbol{L}_B^i = (x_{L_B}^i, y_{L_B}^i)$ は式 (2.15) で表される．

$$
\begin{aligned}
x_{R_A}^i &= L^i \sin\theta_1^i + O_{xf} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{R_A}^i &= L^i \cos\theta_1^i + O_{xf} \sin\theta_2^i - O_{yi} \cos\theta_2^i \\
x_{R_B}^i &= L^i \sin\theta_1^i - O_{xb} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{R_B}^i &= L^i \cos\theta_1^i - O_{xb} \sin\theta_2^i -^i O_{yi} \cos\theta_2^i \\
x_{L_A}^i &= L^i \sin\left(-\theta_1^i - \theta_2^i\right) + O_{xf} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{L_A}^i &= L^i \cos\left(-\theta_1^i - \theta_2^i\right) + O_{xf} \sin\theta_2^i - O_{yi} \cos\theta_2^i \\
x_{L_B}^i &= L^i \sin\left(-\theta_1^i - \theta_2^i\right) - O_{xb} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{L_B}^i &= L^i \cos\left(-\theta_1^i - \theta_2^i\right) - O_{xb} \sin\theta_2^i - O_{yi} \cos\theta_2^i
\end{aligned}
\tag{2.15}
$$

提案するモデルにおける脚先形状は実際のロボットの脚先を包含できる最小の長方形としている．$O_{yi}O_{yo}$ はそれぞれ脚先中心から内側および外側の辺までの長さ，O_{xf}，O_{xb} はそれぞれ脚先中心から前端，後端までの長さである．また，入力列の更新前後で制約式が不連続に変化するのを防ぐために，**??**に示すような左脚座標系から見た左脚内側，右脚座標系から見た右脚内側の形状を表現する関数 $f_f(x)$ を導入する．脚先座標系の y 軸の正の方向に凸で，パラメータによって矩形に近づけられる関数が望ましく，本論文では三つのパラメータ a，b，c をもつ x の 10 乗の関数とした．

$$
f_f(x) = a\,(x-b)^{10} + c \tag{2.16}
$$

パラメータは Levenberg-Marquardt 法による非線形最小二乗推定によりフィッティングを行った．これを用いて，制約条件を式 (2.17) で表現する．

$$
\begin{aligned}
f_f\left(x_{R_A}^i\right) \le y_{R_A}^i \quad , \quad f_f\left(x_{R_B}^i\right) \le y_{R_B}^i, \\
f_f\left(x_{L_A}^i\right) \le y_{L_A}^i \quad , \quad f_f\left(x_{L_B}^i\right) \le y_{L_B}^i
\end{aligned}
\tag{2.17}
$$

2.6.4　二次計画問題への適用

以上の制約条件 $\omega_i(\boldsymbol{q}^0, \boldsymbol{U}_k) \geq 0$ $(i = 0, \ldots, 20k-1)$ は，まとめて $\boldsymbol{\omega}(\boldsymbol{q}^0, \boldsymbol{U}_k) = (\omega_0, \ldots, \omega_{20k-1})^T \geq \boldsymbol{0}$ $(\boldsymbol{\omega} \in R^{20k})$ と表される．これを問題式 (2.7) に適用するため，更新後の制約条件について線形近似を行う．

$$\boldsymbol{\omega}_{\text{next}} = \boldsymbol{\omega}_{\text{now}} + \Omega\, \boldsymbol{\Delta U_k} \geq \boldsymbol{0} \tag{2.18}$$

$\Omega \in R^{20k \times 6k}$ は $\boldsymbol{\omega}_{\text{now}}$ の $\boldsymbol{U}_k$ についてのヤコビ行列 $\Omega = \partial \boldsymbol{\omega}_{\text{now}} / \partial \boldsymbol{U}_k$ である．

2.7　本章のまとめ

本章では，三次元適応のための歩行モデルの拡張および追加の制約条件について説明した．従来の水平方向の脚配置計画のためのモデルに，独立した垂直方向の動作を追加し，脚配置遷移モデルの状態および入力についての次元の拡張を行うことなく，事前に与えられた地形情報に基いて垂直方向の移動量を制限する制約を定式化し追加することで，計算コスト増加を抑えつつ三次元地形への適応を可能にした．次章では，本章で定義した計画問題の解法について，全体のアルゴリズムおよび求解性能の安定化のためのいくつかの手法について述べる．