2Y-04

P2MP 光カメラ通信のためのドローン軌道制御

Drone Trajectory Control for Point-to-multipoint Optical Camera Communication

李 天文[1] 小野寺 幸仁[2] 久野 大介[3] 中山 悠[1]
Tianwen Li Yukito Onodera Daisuke Hisano Yu Nakayama

1 はじめに

近年，LED 等の光源とカメラを用いた光カメラ通信(OCC)が注目されている[1]. OCC では，光源が送信する光信号をカメラが受信するために Line-of-sight(LoS)チャネルが必要になる．1 台のカメラで遠距離の複数ドローンからの光信号を受信する point-to-multipoint(P2MP)-OCC では，光源間干渉の回避が重要な課題となる（図 1）[2]．ただし[2]では，干渉のモデル化は行われている一方で，干渉を回避するためのドローンの軌道制御について十分に検討されていない．そこで本稿では，光源間干渉を回避するためのドローンの分散軌道制御アルゴリズムを提案する．干渉モデルに基づき，撮影画像上でのドローン同士の重なりを回避するように実世界での軌道制御を行う．マルチエージェントシミュレーションにより，提案アルゴリズムを用いて光源間干渉を 100%回避できることを示した．

2 提案アルゴリズム

2.1 概要

提案アルゴリズムでは，各ドローンが目的地へ移動しながら実世界座標と画像上での位置（画像座標と呼ぶ）の相互変換を随時行う．そして画像上での干渉を避けるように実世界で回避・迂回移動をする．

2.2 干渉モデル

本モデルではドローンを半径Rの球体として近似する．実世界座標の原点を受信カメラの位置，Y軸を受信カメラの撮影方向とする．i番目のドローンの実世界座標を(X_i, Y_i, Z_i)とし，これは GPS 等のセンサで取得される．またカメラの焦点距離をf，倍率をMとおく．撮影画像に投影されるドローンの画像座標(x_i, z_i)および半径 r_i は以下のように計算される．

$$(x_i, z_i, r_i) = \left(\frac{fMX_i}{Y_i}, \frac{fMZ_i}{Y_i}, \frac{fMR_i}{Y_i}\right) \tag{1}$$

ドローンの配置に関する制約条件としては，[3]で提案された近似モデルを採用する．これはカメラと光源の距離が比較的遠い場合を想定したシンプルなモデルである．また，2 次元イメージセンササイズを(ϵ_w, ϵ_h)とすると，カメラの撮影範囲は以下の式で表される．

$$\begin{aligned} -\left(\frac{\epsilon_w}{2} - r_i\right)\frac{Y_i}{fM} \le X_i \le \left(\frac{\epsilon_w}{2} - r_i\right)\frac{Y_i}{fM} \\ -\left(\frac{\epsilon_h}{2} - r_i\right)\frac{Y_i}{fM} \le Z_i \le \left(\frac{\epsilon_h}{2} - r_i\right)\frac{Y_i}{fM} \end{aligned} \tag{2}$$

2.3 アルゴリズム

提案アルゴリズムの目的は，カメラの撮影範囲と干渉に関する制約条件を満たしながらドローンの移動距離の増大を抑制することである．ドローンは最短経路で目的地へ向けて移動しながら，周期的に(X_i, Y_i, Z_i) を取得し，画像座標系での探索を行う．具体的な手順として，「前方探索」と「全方位探索」の 2 段階の探索を行う (図 2).

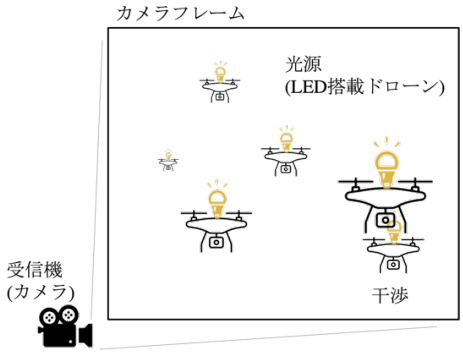


図 1 P2MP をする際の撮影画像上での干渉

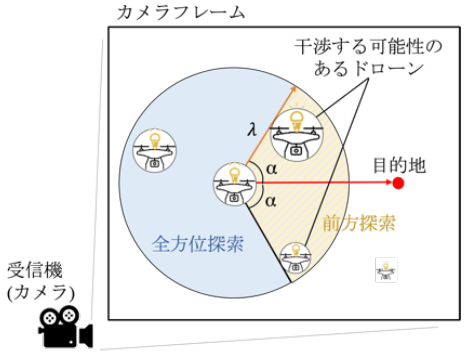


図 2 ドローンの探索方法

以下では，提案アルゴリズムの詳細について順に説明していく．

1) 速度ベクトル

ドローンiの速度ベクトルを$\vec{V_i}$（$|\vec{V_i}| = 1$），スピードをk_iとする．速度ベクトルは現在地から目的地への方向ベクトルである．速度更新の時間間隔をτとおけば，τ後のドローンの座標は以下の式で表される．

$$(X_{i,t+\tau}, Y_{i,t+\tau}, Z_{i,t+\tau}) = (X_{i,t}, Y_{i,t}, Z_{i,t}) + k_i\tau\vec{V_i} \tag{3}$$

2) 前方探索

撮影画像上での速度ベクトルを $\vec{v_i}$ （$|\vec{v_i}| = 1$）とする．前方探索では$\vec{v_i}$から$\angle\alpha$以内を探索する．探索範囲内にあるドローンへの方向ベクトルを $\vec{Y_{ij}}$ とし，$\vec{Y_{ij}}$ と $\vec{v_i}$ とのなす角を$\angle\beta_{ij}$とする．撮影画像上におけるドローンiとドローンjのユークリッド距離をδ_{ij}とすると半径を考慮したドローン同士の距離は以下のように示される．

$$s_{ij} = \delta_{ij} - r_i - r_j \tag{4}$$

$s_{ij} < p$ を満たすドローンの集合をNとする．pは閾値である．この集合Nの大きさによってドローンの次の動作が決定される(図 3)．回避する場合，回避ベクトル$\vec{c_i}$ は以下のように決定される．

$$\vec{c_i} = (c_x, c_z) = -\vec{Y_{ij}} \tag{5}$$

[1] 東京農工大学情報工学部 [2] 東京農工大学大学院工学府情報工学専攻
Department of Computer and Information Sciences,
Tokyo University of Agriculture and Technology
[3] 大阪大学大学院工学研究科
Graduate School of Engineering, Osaka University

迂回する場合，迂回先の座標は次のように決定される．

$$(X_{i,tmp}, Y_{i,tmp}, Z_{i,tmp}) = (X_i, Y_i, Z_i) + \vec{\sigma}(2r_{max} + p) \quad (6)$$

r_{max}は撮影画像上での最大のドローンサイズであり，$\vec{\sigma}$は $\vec{v_i}$ から$\angle\alpha$以上$\angle 2\alpha$以下の範囲内におさまる単位ベクトルである．$|\vec{V_i}|$は迂回先の目的地を使って更新される

3） 全方位探索

撮影画像上で他のドローンと距離を保つために全方位探索を行う．全方位を探索し，$s_{ij} < \eta p$ $(0 < \eta < 1)$ を満たすドローンがいるとき回避ベクトル $\vec{c_i}$ は次のように更新される．

$$\vec{c_i} = \eta\vec{c_i} - \sum_{j \in N'} \vec{r_{ij}} \quad (7)$$

全方位探索の重みが前方探索よりも大きいのは，探索範囲が短いためである．

4） 速度更新

回避ベクトルにより速度ベクトルを更新する．回避ベクトルは撮影画像上のベクトルなので実世界座標系に変換する．

$$\vec{C_i} = (C_X, C_Y, C_Z) = \left(\frac{c_x Y_i}{f}, 0, \frac{c_z Y_i}{f}\right) \quad (8)$$

速度ベクトルの更新は以下のように行う．

$$\vec{V_i} = \vec{V_i} + u\vec{C_i} \quad (9)$$

uは$\vec{C_i}$の大きさを調整する変数であり，次のようである．

$$u = \left(\frac{\sqrt{1 - \vec{V_Y}^2}}{\sqrt{C_X^2 + C_Z^2}} - 1\right)\left(\frac{\vec{V_X}}{C_X}, 0, \frac{\vec{V_Z}}{C_Z}\right) \quad (10)$$

次に速度の更新を行う．撮影画像上での最大制動距離λは以下のように決定される．

$$\lambda = r_i + r_{max} + \frac{k_{max}^2}{2a_i}\frac{fM}{Y_i} \quad (11)$$

a_iはドローンiの加速度を示す．ドローンは全方位探索と同じ方法で$s_{ij} < \lambda$ を満たすドローンを探索する．もしドローンがいない場合はa_iだけ加速し，ドローンがいる場合はa_iだけ減速する．

5） 到着判定

ドローンが動くたびに目的地到着判定を行う．もし仮の目的地であれば，元の目的地に更新し直す．

3 シミュレーション

3.1 シミュレーション環境

提案する軌道制御アルゴリズムの性能を，Python の Multi Agent Simulation (MAS)モジュールである Mesa を用いて評価した．フルサイズカメラを想定し，2 次元イメージセンササイズを 24mm×36mm，焦点距離を 35mm とした．倍率は 3 倍とした．ドローンは半径 25 cm の球体とし，最大速度は 36km/h である．初期座標と目的地座標は式(2)を満たし，50m< Y_i<200m の範囲でランダムに決定した．提案アルゴリズムにおける更新間隔$\tau = 0.1$秒に設定した．提案アルゴリズムを以下の 2 つのアルゴリズムと比較した．

1) straight: 各ドローンは目的地まで直進する．

2) wait: 2 機のドローンが撮影画像上で接近した場合，奥にあるドローンは停止し，もう 1 機は干渉を回避する[3].

シミュレーションはドローンの数を変えながらそれぞれ 1000 回繰り返し，各シミュレーション時間を最大 200 秒までとした．

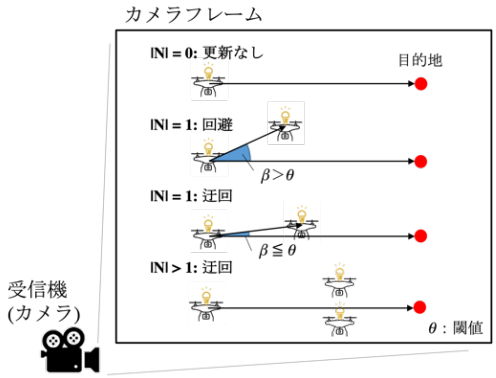


図 3　条件分岐

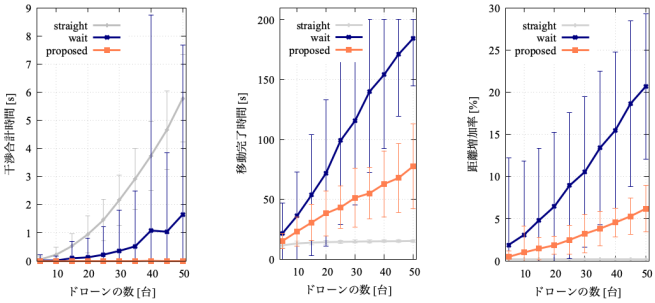
3.2 シミュレーション結果

シミュレーション結果を図 4 に示す．

図 4(a)は撮影画像上での干渉の総時間を示す．提案アルゴリズムでは，軌道制御により光源間干渉を回避できた．

図 4(b)は全てのドローンが目的地に到着した時刻を示す．提案アルゴリズムでは効率的な回避・迂回により時間増加を抑制できた．

図 4(c)はドローンの移動距離が増加する様子を示す．この値は最短経路からの増加率で表される．提案アルゴリズムでは，増加率を 10%未満に抑えることができた．



(a) 干渉合計時間　(b) 移動完了時間　(c) 距離増加率

図 4　シミュレーション結果

4 結論

本稿では，ドローンを用いた P2MP-OCC のための分散軌道制御アルゴリズムを提案した．提案アルゴリズムにより，各ドローンは他のドローンとの光源間干渉を回避し，LoS リンクを確保しながら効率的に目的地に移動できる．干渉回避の性能は撮影画像上の干渉モデルによって理論的に保証されている．提案アルゴリズムの有用性はマルチエージェントシミュレーションにより確認した．今後はドローンを用いた実環境実験で提案アルゴリズムを評価していく．

謝辞

本研究の一部は，JST ACT-I(JPMJPR18UL)，さきがけ(JPMJPR2137)，および GMO 財団の支援を受けて行われた．

参考文献

[1] N. T. Le, et al. Signal Processing: Image Communication,2017.
[2] Y. Onodera, et al. IEEE GLOBECOM, 2021.
[3] Y. Onodera, et al. IEEE CCNC, 2022.