

小型 UAV による自律型 SLAM システムの実証研究

Verification study of autonomous SLAM system using small UAV

酒井 研斗 堀 浩一
Kento Sakai Koichi Hori

東京大学 大学院工学系研究科 航空宇宙工学専攻

Department of Aeronautics and Astronautics, School of Engineering, The University of Tokyo

In this study, we aim to achieve "autonomous SLAM" by using small Unmanned Aerial Vehicle (UAV). That is an autonomous explore system of unknown environment without manual operation. Apply the flying rule suit to the constraints in SLAM by the feature points matching and conducted real machine experiment. A result, we achieved robust fully autonomous search in a broad indoor, but failed to advanced search, such as examine all over the unexplored area. Therefore, this study showed that autonomous SLAM using a small UAV is realizable.

1. 研究背景および目的

近年、航空分野において小型無人航空機（小型 UAV）、特にクアッドローター（Quadrotor）は垂直離着陸、ホバリングや超低速飛行が可能な点から注目されている機種のひとつである。一方、ロボットの知能化、コンピュータビジョンの分野において、SLAM(Simultaneous Localization and Mapping)も近年盛んな研究テーマの一つである。SLAMとは、カメラなどを用いて自己位置推定と周囲環境のマッピングを同時にを行うことを言い、ルンバ [iro] などが SLAM を行う自律型ロボットとして知られる。その機動性から UAV の SLAM への適用が注目されている。UAV による SLAM は、安定性や自己位置推定の精度の問題はあるが、飛行しているために障害物や段差に強く、より詳細な地図を得られることが期待される。UAV による SLAM の中でも特に、人間による操作を必要とせず自ら意思決定を行う "自律型 SLAM" を実現することは、将来的に、環境を選ばずミッションをこなす知的 UAV を実現する上で必要不可欠であり、UAV の研究開発において重要であると考える。

それゆえ本研究では、次の 2 点を満たすような自律型 SLAM システムを UAV を用いて構築する。

1. システムは、周囲環境・自身の状態をリアルタイムで認識する
2. システムは、認識した情報にもとづいて、マニュアル操作なしに探索行動を行う

本稿では、まず提案するシステム作成の上で利用した 2 つのパッケージ、PTAM と tum_ardrone の解説を行う。次にその特性を考慮した探索ルーチンを提案し、システム全体の構成を確認する。続けて実機実験の結果を示し、最後にシステムの性能評価と改善すべき点について検討する。

2. 環境地図作成とリアルタイム自己位置推定

2.1 特徴点地図の作成と自己位置推定-PTAM

本研究では、環境地図作成と自己位置推定は PTAM[Klein 07] によって実現されている。PTAM は

連絡先: 酒井研斗、東京大学 大学院工学系研究科 航空宇宙工学専攻、堀研究室, sakai@ailab.t.u-tokyo.ac.jp

正式には Parallel tracking and mapping と称される Visual SLAM(カメラなど画像による SLAM) ライブリである。

2.1.1 環境地図表現

PTAM の環境地図は 3 次元慣性座標系 \mathcal{W} 中に M 個の特徴点を保持している。特徴点はエッジなどを特徴量と呼ばれる高次元ベクトルで表現したものなので、2 次元画像を局所的に切り出したものと言い換えられ、同様に環境地図は、カメラ画像をごく小さな"布"として 3 次元空間中に貼り付けたものと考えることが出来る (1)。環境地図はまた、 N 個の Keyframe

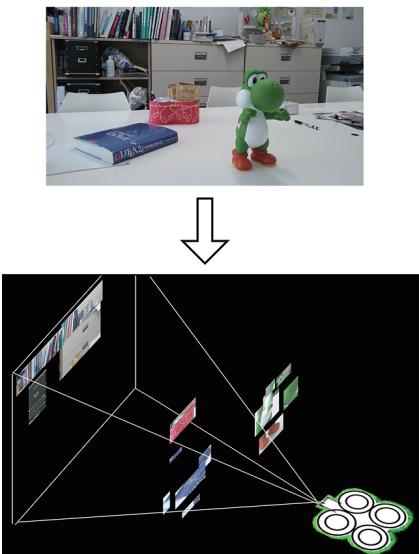


図 1: PTAM の環境地図イメージ。実際は特徴点なのでより細分化されている

を保持している。これらはある時刻でのカメラ画像のスナップショットであり、各々がカメラ中心の座標系 \mathcal{K}_i と、PTAM の慣性座標系との並進、移動変換 $E_{\mathcal{K}_i \mathcal{W}}$ を持つ。

2.1.2 Tracking-自己位置推定

PTAM のトラッキングはマッピングとは独立したスレッドで、毎フレームごとにカメラの姿勢を求めていく。

カメラ画像からは FAST-10 特徴点抽出 [Rosten 06] によっ

て特徴点が抽出され、また前準備として、フレームごとの姿勢変化を元に、減衰運動モデルによる粗い姿勢推定 E_{CW} を得る。次に E_{CW} を補正する並進回転変換を Exponential Map を用いて得られる 6 次元ベクトル [Varadarajan 74] μ で $\exp(\mu)$ と表す。環境地図中の特徴点は、これらの合成変換 $\exp(\mu)E_{CW}$ とカメラ投影モデルを用いてカメラの 2 次元画像中へ投影される。この補正值 μ を変化させた時に、カメラ画像中の特徴点と、投影された環境中の特徴点の誤差の総和が最も小さくなるような μ' を求める。

こうして得られた合成変換 $E'_{CW} = \exp(\mu')E_{CW}$ がトラッキングの出力であり、そのフレームでのカメラ姿勢として利用される。

2.1.3 Mapping-環境地図の作成

PTAM の環境地図は、まず [Stewenius 06] の 5 点アルゴリズムによって初期化される。カメラを並進移動を挟んだ 2 点間で命令を与えると 2 つのカメラ位置間の変換が計算され、マッチングした特徴点を三角法によって投影し、初期マップが作成される。

初期化後は、「Tracking の精度が good 以上」「最後に Keyframe が追加されてから 20 フレーム以上経過」「カメラが最も近い Keyframe 撮影位置からある距離以上離れている」の条件が満たされた時、Keyframe が追加される。追加される Keyframe は撮影位置が最も近い Keyframe を比較画像として対応する特徴点を探す。これで対応が取れた特徴点は、やはり三角法によって地図上にマッピングされる。

このように、カメラがトラッキングを維持しながら移動していくことで、順次 Keyframe が追加され、環境地図は大きくなっていく。

2.2 自己位置推定のリアルタイム性・ロバスト性の向上-tum_ardrone

PTAM の自己位置推定の頻度はカメラ画像の送信頻度（一般的な単眼カメラの場合、30fps 程度）である。また、トラッキングに失敗している間は意味のある推定が行えない。そのため、PTAM の自己位置推定を高いリアルタイム性が要求される UAV の状態量としてそのまま用いることは出来ない。

本研究ではこれを解決し UAV での SLAM を実現させた tum_ardrone[Engel 11] パッケージを利用している。

状態量推定のリアルタイム性、ロバスト性を向上させるため、tum_ardrone では拡張カルマンフィルタ (EKF) を用いている。状態量は以下の通りである。

$$\mathbf{x}_t := (x_t, y_t, z_t, \dot{x}_t, \dot{y}_t, \dot{z}_t, \Phi_t, \Theta_t, \Psi_t, \dot{\Psi}_t)^T \in \mathbb{R}^{10} \quad (1)$$

ここで、 (x_t, y_t, z_t) は機体重心座標 (m)、 $(\dot{x}_t, \dot{y}_t, \dot{z}_t)$ は機体速度 (m/s)、 $(\Phi_t, \Theta_t, \Psi_t)$ は機体のロール、ピッチ、ヨー角 (degree)、そして $\dot{\Psi}_t$ はヨー角速度 (degree/s) を表す。

この EKF は、IMU と高度計から得られる 200Hz 程度のオドメトリ観測モデル (主に $(\dot{x}_t, \dot{y}_t, \dot{z}_t, \Phi_t, \Theta_t, \Psi_t)$)、PTAM の画像トラッキングから得られる 30Hz 程度の画像観測モデル $((x_t, y_t, z_t, \Phi_t, \Theta_t, \Psi_t))$ と、二種類のモデルによって、観測値が得られた時だけ更新を行う。この EKF がトラッキングの失敗にロバストかつリアルタイム性の高い自己位置推定を実現している。

また、計算能力の関係から多くの処理は無線によってオフボード PC で行われるが、この時各観測値が EKF に到達するまでの観測遅れ、また制御命令が機体に届くまでに制御遅れが存在する。そこで EKF では観測値、制御命令、状態量にタイ

ムスタンプをつけてバッファリングし、観測遅れの分だけ過去から始め、制御遅れの分だけ未来までを予測計算している。

この制御遅れ分未来の状態量と目標値を、更に PID 制御にかけ、制御命令を機体に送信している。

3. 環境・状態の分析と行動決定ルーチン

ここまでで、機体の座標姿勢、周囲環境を認識しながら、UAV に目標位置を定めた飛行が可能である。

本研究では自律型 SLAM を目指すため、この飛行目標位置を環境・状態に応じて動的に決定するルーチンを作成した。このルーチンを、便宜上 SearchPathFinder と呼ぶ。SearchPathFinder は自律的に探索を行い、異常時であってもマニュアル操作を受けられないため、行動方針を、優先順位をつけて以下のように設定した。

1. 落下することなく飛行し続ける
2. 自己位置推定を保って飛行し続ける
3. より広い方位方向を視野に収める

ここでは、これらを満たすよう実装した SearchPathFinder の行動ルーチンについて述べる。

3.1 PTAM の特性を考慮した繰り返し探索

2.1.1 で述べた通り、PTAM の地図はいわば、カメラ画像を小さく切り出して空中に貼り付けたようなものである。そのため、原理的に横方向からのマッチングが難しく、カメラ画像の射影変化が大きい旋回行動では、トラッキングが失敗しやすい。逆に前後移動ではカメラ画像中の特徴点の消失は少な

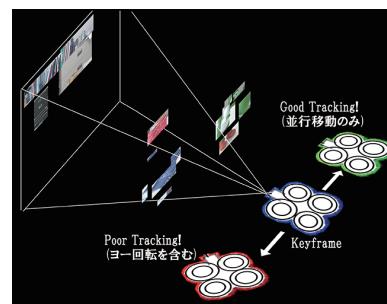


図 2: トラッキングの成功しやすい変換、失敗しやすい変換

く、画像の射影変化も小さいため、トラッキングを安定させつつ多くの地図情報を得ることが出来る。こうした理由から、SearchPathFinder は次の繰り返し行動で探索を行う。

1. 「0.5m 後退」を 4 回繰り返す
2. 「10° 右旋回」を 4 回繰り返す
3. 「0.5m 前進」を 4 回繰り返す
4. 「0.2m 左並進」を 4 回繰り返す
5. 1. に戻る

これが推定誤差や外乱なく達成された場合の経路は 3 のようになる。先に後退し壁面からの距離を大きくとることで旋回時の画像変化を小さくし、旋回時のトラッキングの安定化を図る。また、全体として初期化位置からは大きく動かさずに周囲を見渡すような経路になっている。ただし、この経路の達成自体は SearchPathFinder の目的ではないことに注意したい。

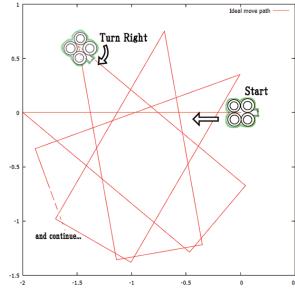


図 3: 誤差、ノイズのない場合の予想経路

3.2 モード切り替えによるロバスト探索

SearchPathFinder はまた、4 のような段階的状態遷移を行う。Stable モードでは前節の繰り返し探索を行う。ここでト

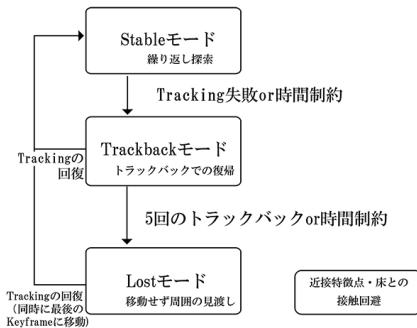


図 4: SearchPathFinder の状態遷移

ラッキングが失われると、Trackback モードに移行し、保存された行動履歴を遡ってトラッキング可能な状態に戻ろうとする。これは EKF がトラッキング失敗時でも推定を続けることで可能となっている。それでもトラッキングが回復しない場合、Lost モードに移行する。ここでは EKF の座標は時間経過により信頼出来ないとして移動はせず、その場で大きく旋回してトラッキングの回復を待つ。

4. 機体及びその他の利用システム

機体は Parrot 社の AR.Drone 2.0 を使用した。計算性能の制約により、機体はオフボード PC にセンサ情報を送り、制御命令を受け取るのみである。ここまで内容を統合した本研究のシステム構成全体は 5 になる。

5. 実験と評価

5.1 広い屋内での実験

実験は横幅 7.5m、奥行き 14.4m、天井高さ 2.7m の、大きい教室程度の広い部屋で行った。部屋の中央付近の高さ約 1m、机程度の高さから飛行を始めた。飛行中、トラッキング自体に失敗することはあってもトランクバックなどによって復帰することで想定の範囲で継続して飛行を続けた。最終的に、飛行時間は 9 分 33 秒、観測方位角の大きさ（開始から停止までに Drone 自身のが観測した方位角度）は 360° 全方向となっ

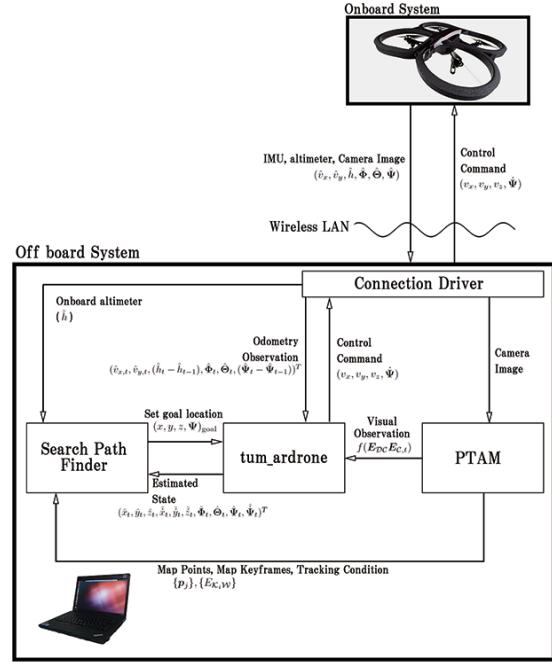


図 5: 本研究のシステム構成図

た。飛行開始から停止させるまで、マニュアル操作は行っていない。

その時の飛行経路は 6 となり、カメラ画像による 3 次元再構成の結果は 7、8 となった。この再構成図は PTAM のアルゴリズムとの共通点から、システムが認識していた環境地図とほぼ同じと言える。

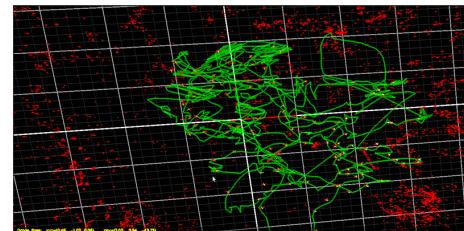


図 6: 飛行経路 1, 上面図

また、同じ条件にて飛行を行ったところ、数分間に渡って飛行し 180 以上の観測方位角を得られるものが半数以上であった。

6. 結論

6.1 評価

飛行中、トランクバックなどによって探索が続行できなくなる状態を回避したことを踏まえると、飛行時間、観測範囲、そして得られた飛行経路はシステムが自律的な探索を行ったと言え、またそうであれば、小型 UAV による自律型 SLAM システムを実証するという本研究の目的は達成されたと考える。

ここで 1 節で述べた 2 点の目標について考える。2 点のうち、周囲環境・自身の状態の認識という点については十分である。一方で、マニュアル操作なしの探索行動については、探

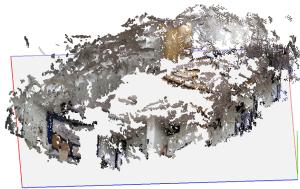


図 7: 3 次元再構成, 前面右側面上方からの鳥瞰図

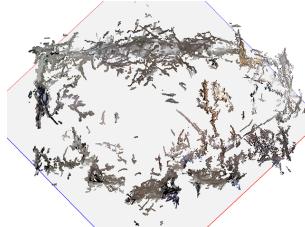


図 8: 上面図

索は出来ているが、方針はヒューリスティックによるものであり、知的な行動を取ってはいない。この点において本研究は自律型 SLAM の最低限の要件を実証したのみであり、大いに発展の余地がある。

6.2 自律型 Visual-SLAM におけるトラックバックの副次的接触回避効果

実験において、壁面に近づいた機体が接触する前に引き返すという現象が多く見られた。Visual-SLAM では壁面に近づくとカメラ画像は乱れが大きくなり、視野も狭くなるため、トラッキングに失敗しやすい。こうして Trackback モードに入った機体が来た軌跡を引き返すという一連の過程が接触回避の効果を持つことになっていた。これは、PTAM に限らず、画像トラッキングを行う自律型 SLAM では同様に起こりうる。

つまり、自律型 Visual-SLAM では、異常時トラックバックを行うことで正面方向からの接触をある程度抑えることが出来ることが分かった。

6.3 今後の課題

6.3.1 特徴点ベース Visual-SLAM の本質的問題

2.1.1 で述べた通り、特徴点ベースの環境地図はカメラ画像を 3 次元空間へ切り貼りしているようなものであり、原理的にマッチング出来る角度に限界がある。

これを克服するには環境地図を 3D メッシュとして記憶し、全方向からの視認結果を表現出来るようにするといった方法が考えられる。

6.3.2 環境地図のスケーラビリティの問題

実験において、探索範囲の増加に伴ってトラッキングの性能が落ちるという現象が確認された。扱う地図情報が増えることでマッチング対象となる情報が増えるため起こる現象である。

PTAMM[Castle 08] では環境地図を複数持った上で現在の地図を切り替えるという方法でこれに対処する試みがなされている。このように、何らかの方法で環境地図全体をマッチングに使用する領域とそれ以外に分けることで、この問題に対処出来ると考えられる。

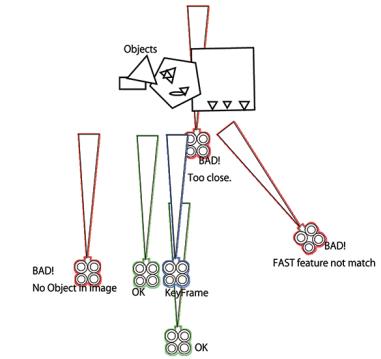


図 9: 特徴点ベース SLAM のトラッキング特性概略

参考文献

- [Castle 08] Castle, R., Klein, G., and Murray, D. W.: Video-rate localization in multiple maps for wearable augmented reality, in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pp. 15–22IEEE (2008)
- [Engel 11] Engel, J.: Autonomous Camera-Based Navigation of a Quadrocopter, Master's thesis, Technical University Munich, Germany (2011)
- [iro] iRobot, <http://www.irobot-jp.com/>
- [Klein 07] Klein, G. and Murray, D.: Parallel tracking and mapping for small AR workspaces, in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234IEEE (2007)
- [Rosten 06] Rosten, E. and Drummond, T.: Machine learning for high-speed corner detection, in *Computer Vision–ECCV 2006*, pp. 430–443, Springer (2006)
- [Stewenius 06] Stewenius, H., Engels, C., and Nistér, D.: Recent developments on direct relative orientation, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 60, No. 4, pp. 284–294 (2006)
- [Varadarajan 74] Varadarajan, V. S.: Lie groups, Lie algebras, and their representations (1974)