

自律巡視ロボットのためのコンピューティングプラットフォーム

プロジェクト代表者：理工学部・教授 泉 知論

我々は、寺社の人災・獣害への対策として、監視カメラによる見守りの高度化の研究を進めている。これまで、潜在的脅威となる対象を自動で認識し発報する監視カメラシステムの開発を目指し、人物の検出および危険度の判定と発報、不審行動の検出、害獣の自動検出などの研究開発を進めてきた。一方で、対象範囲を隅々まで見守るには、多数の監視カメラを配置する必要があり、またそれらのための配線を張り巡らせる必要がある。そこで、より能動的な対策として、監視カメラの自律巡視ロボット化の研究を進めている。

カメラにより外界を認識し、見守り対象領域を巡視する小型の自律移動ロボットの実現には、小型軽量低消費エネルギーな高効率なコンピューティングプラットフォームが必要になる。我々は再構成可能ハードウェアを活用した高効率な組込み向けコンピューティングプラットフォームの研究を進める。2020年度は、特に自律的な経路探索について、障害物を認識し回避する機能の効果的な実装法を検討した。

近年の経路計画の研究において、生成経路の質と計算量のバランスを取りやすいランダムサンプリングに基づく手法が提案されている^[1]。その中で Informed-RRT*^[2] は中継候補点のサンプリングを改善の可能性がある超楕円内に限定することにより、さらに計算効率向上をはかる手法である。図1に経路探索の例を示す。出発点 X_{start} から目標点 X_{goal} に向かう経路を探索する際、既知最短経路の距離から探索範囲を破線の楕円に限定する。さらに FPGA

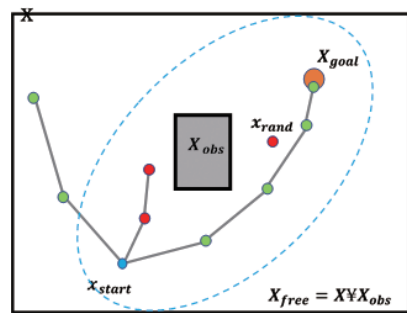


図1：Informed-RRT* による経路探索

によるハードウェア化の研究^[3]も進められている。Informed-RRT* はサンプリング範囲を限定できる一方で、直接的な超楕円内からの一様サンプリングには三角関数や n 乗根の計算を必要とし、計算能力に限られる環境下には計算負荷が大きく、また、再構成可能ハードウェアによる実装にも向かない。一方で、超楕円を含む矩形から一様サンプリングして棄却する方法（以下、棄却サンプリング）では、棄却の無駄はあるが計算は単純であり、ハードウェア化もしやすい。棄却率は対象とする状態空間の次元に依存し、計算負荷は乱数生成アルゴリズムにも依存する。棄却サンプリングの方が結果として計算効率が良くなる可能性がある。本研究では、単位超球内サンプリングにおいて直接サンプリングと棄却サンプリングでの乱数の生成速度の比較を行い、棄却サンプリングを用いることで性能が向上する条件を検討する。

まず、基礎評価として組込みプロセッサを用いてサンプリングに要する計算時間の測定実験を行った。結果を図2、図3に示す。棄却率は矩形と楕円の面積比（超立体の体積比）にほぼ等しく、乱数の生成速度は棄却サンプリングを用いることで2次元空間では約4.48倍、3次元空間では約3.35倍、4次元空間では約1.7倍向上した。ここで言う状態空間とは、ロボットが移

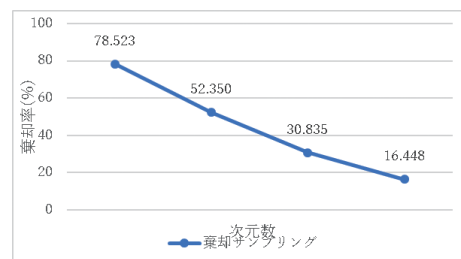


図2：棄却サンプリングの棄却率

動する実空間にロボットの状態（ステアリング角度やアームの各軸の角度など）も加えた高次元の空間である。平面上を低速で移動する差動二輪の自律移動ロボットであれば、2次元から多くても4次元程度であり、棄却サンプリングの方が優位になる。

また、小型自律移動ロボットでのFPGAによる高速化に適した乱数生成アルゴリズムを検討した。標準的なソフトウェアライブラリで多く採用される線形合同法、高性能なアルゴリズムとして知られるメルセンヌ・ツイスタ、デジタル回路としての実装に向く線形帰還シフトレジスタについて、FPGAでの回路規模と処理速度、ソフトウェア上での規定回数の乱数生成速度を評価する。結果として、FPGAでの回路規模と処理速度は線形帰還シフトレジスタが最も小さく、ソフトウェア上での処理速度は、線形合同法と線形帰還シフトレジスタはほぼ等しく、メルセンヌ・ツイスタは2つの約10倍遅かった。定性的には、乱数には暗号学的安全性は不要であり、比較的単純な古典的疑似乱数生成アルゴリズムで十分と考えられる。線形帰還シフトレジスタが妥当な選択である。

Informed-RRT* プログラム^[4]に線形帰還シフトレジスタを用いた棄却サンプリングを組み込み生成される経路の精度を評価した。図4に経路探索の様子を示す。提案手法でも、メルセンヌ・ツイスタを用いた超楕円サンプリングによるものと比べ生成される経路の距離に劣化は見られなかった。

今後、線形帰還シフトレジスタを用いた棄却サンプリングを組み込んだInformed-RRT*について、ハードウェア化を検討していく。

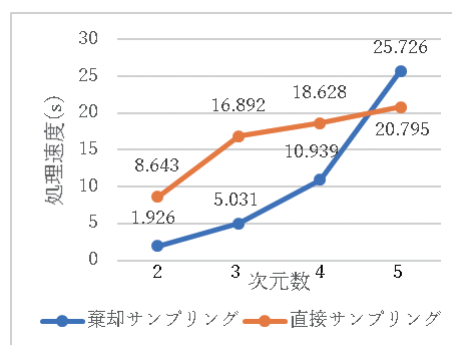


図3：サンプリングに要する時間

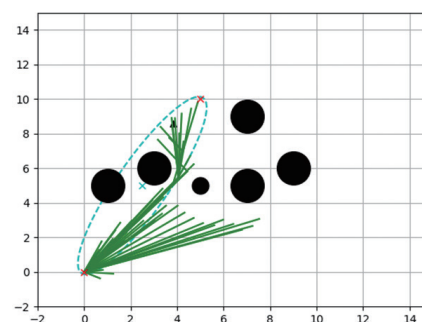


図4：Informed-RRT* に提案手法を組み込んだ経路探索の様子

参考文献

- [1] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", International Journal of Robotics Research, vol. 30, no. 7, pp. 846-894, 2011.
- [2] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: optimal samplingbased path planning focused via direct sampling of an admissible ellipsoidal heuristic", in Proceedings of International Conference on Intelligent Robots and Systems (IROS), pp. 2997-3004, 2014.
- [3] S. Xiao, N. Bergmann, and A. Postula, "Parallel RRT* architecture design for motion planning", in Proceedings of 27th International Conference on Field Programmable Logic and Applications (FPL2017), pp. 1-4, 2017.
- [4] Atsushi Sakai, "PythonRobotics: Python sample codes for robotics algorithm", <https://github.com/AtsushiSakai/PythonRobotics>.