

RISC-V エミュレータを用いたオンボード画像航法システム における参照画像生成プログラムの検証

高島 浩司† 伊与田 健敏† 津田 雄一‡

創価大学† 宇宙航空開発機構‡

1. はじめに

2020 年 12 月, 探査機はやぶさ 2 は小惑星リュウグウからサンプルを地球に届け新たな旅へ出発をしている. 地球とリュウグウ間は約 3 億 4000 万 km で, はやぶさ 2 が直径 900m のリュウグウに到着するためには正確な推定位置情報が必要不可欠である. はやぶさ 2 の推定位置は RARR, DDOR 手法を用いることにより誤差が約数 km となる. しかし, リュウグウの推定位置にも誤差があり, 約 200km である. リュウグウの推定位置誤差を修正するための手法として画像航法[1], 光学航法[2]がある. この 2 つは, 地上局で処理を行うため, 送受信時間 50 分と計算時間 10 分を合わせて約 1 時間かかる. さらに, 遠方天体で探査機のホームポジションがリュウグウの場合より高い場合, タッチダウンするまでのプロセス時間は長くなる. さらに, 探査機がホームポジションから降下する際, 目標天体の引力を受けるためリアルタイム処理の優先度, 重要性は高まる. 私たちは探査機内のシステムとして画像航法システムを組み込むことに着目した.

本研究では, 画像航法システムを本学で研究が行われた画像マッチング手法であるベクトル符号相関法(VCC)[3]に置き換えることを検討している. 前段階として, VCC で必要となる参照画像を RISC-V コアを用いて生成することを目指す.

2. オンボード画像航法システムの概要

オンボード画像航法システムの概要を図 1 に示す. STL ファイルにある 3D モデルの頂点, 法線データより生成される参照画像と探査機搭載カメラで撮像された入力画像のピクセルごとのベクトル符号(4bit)を求める. 2つのベクトル符号の相関値を加算し続け, 全ピクセルの相関値計算が終了してから, テンプレートマッチング処理を行う. 処理結果より, 目標天体の正確な推定位置を補正できる.

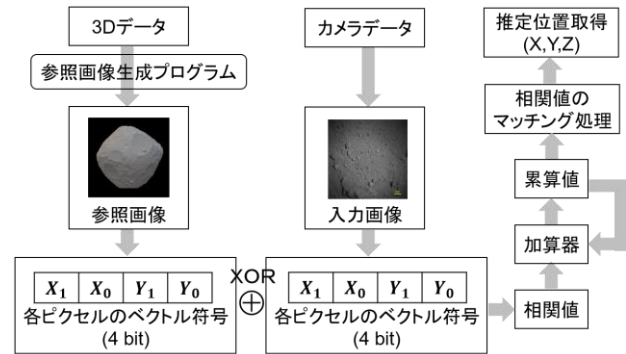


図1 オンボード画像航法システムの概要

テンプレートマッチング手法は VCC を検討している. VCC は, ハードウェアとして確立されていて, 高速なマッチング処理が可能である[4]. 対応するピクセルのベクトル符号の相関値よりマッチング処理を行う.

3. 参照画像生成方法

参照画像は OpenGL ライブラリと呼ばれる 3D グラフィックスに特化したインターフェースを用いることで生成される. OpenGL は GPU としてハードウェア化されているが, 消費電力が多く, 資源が少ない探査機内のシステムとして組み込むことは不可能である. そのため, OpenGL で使われる GL 関数を解析し, 組み込みプログラムに変換する必要がある.

参照画像生成の流れは3つの処理に分けることができる. 1つ目の処理は, STL ファイルからデータを読み出す処理である. 今回使用した STL ファイルは2種類あり, バイナリ形式で非圧縮の約 27M バイトと4分の1に圧縮した約 6.5M バイトのファイルである. 圧縮ファイルを使用する理由は, メモリのオーバーフローを防ぐためである. 2つ目の処理は, 読み込んだ各頂点を GL 関数に渡す処理である. GL 関数では, 各三角形の頂点ごとに外積と法線を求めている. 3つ目の処理は描画である. GL 関数によって求めたデータをもとに 2D 画像を生成する. この 2D 画像が参照画像となる.

小橋ら[5]によって GL 関数内での詳しい処理の解析は終了しており, OpenGL を使用せず, gcc

An examination reference image generation program for onboard image navigation system using RISC-V emulator

†Koji Takashima †Taketoshi Iyota ‡Yuichi Tsuda
†Soka University ‡JAXA

コンパイラ上で参照画像は正常に生成されている。

4. 検証実験

4.1. 実験概要

VCC は FPGA 内の回路として組み込まれており、高速なマッチング処理が可能である。しかし、参照画像生成プログラムが動作する回路、CPU は未完成である。CPU コアはオープンソースでフリーライセンスな RISC-V[6]を採用し、開発を進めている。

そこで、既存の RISC-V エミュレータを用いて参照画像生成プログラムの動作検証を行う。非圧縮 STL ファイルと圧縮 STL ファイルそれぞれで実験を行った。RISC-V エミュレータは QEMU のユーザーモードエミュレーションを用いて起動した。実行コマンドは、`riscv-gnu-toolchain`[7]にある `riscv64-unknown-elf-run` コマンドを用いた。

4.2. 実験結果

非圧縮 STL ファイルを用いた場合、セグメンテーションエラーより参照画像を生成することができなかった。これは、仮想メモリが 64M バイトであるにも関わらず、読み込むための配列や描画するための配列に約 75M バイト費やしているからだと考えられる。圧縮 STL ファイルを用いた場合、参照画像を生成することができた。

4.3. 考察

今回の実験で生成した参照画像の評価をするために、非圧縮 STL ファイルを用いた場合の参照画像と比べて変化があるか確認する必要がある。gcc コンパイラを用いて非圧縮 STL ファイルから参照画像を生成した。図 2 に生成した参照画像の比較を示す。リュウグウ全体を比較すると変化がないように思うが、一部拡大を比較すると頂点とポリゴンが少し荒くなっていることがわかった。これは、圧縮 STL ファイルの頂点数が非圧縮 STL ファイルの頂点数より少ないためだと考えられる。

5. まとめ

さらなる遠方探査を行う際、通信時間の莫大な増加に伴い、画像航法システムを探査機内に組み込むことは重要である。私たちは、オンボード画像航法システムのテンプレートマッチングの際に必要な参照画像生成に着目し、組み込

みシステムで動作するか RISC-V エミュレータを用いてシミュレーションを行った。結果として、非圧縮ファイルは参照画像生成できず、圧縮ファイルは動作をしたが参照画像が荒くなってしまった。

今後は、RISC-V エミュレータを用いて生成した参照画像と仮想的に見立てた入力画像を用いて VCC を行い、正しく処理が来ているのかシミュレーションを行っていきたい。

参考文献

- [1] G. Ohira, S. Kashioka, Y. Takao, M. Kadokura, T. Iyota and Y. Tsuda: *On-board Processing of Vector Code Correlation Algorithm for Image-based Navigation System*, The 28th Workshop on JAXA Astrodynamics and Flight Mechanics, SA6000135065, C-12 (2018)
- [2] Y. Takao, Y. Mimasu and Y. Tsuda: *Optical Navigation for Hayabusa2 Approaching Phase*; The 28th Workshop on JAXA Astrodynamics and Flight Mechanics, SA6000135032, B-6 (2018)
- [3] 小山雄輔, 崔龍雲, 伊与田健敏, 久保田謙: ベクトル符号相関を用いたテンプレートマッチングの高速化; FA2-09-5, p.785-788, <https://doi.org/10.11511/jacc.46.0.785.0> (2004)
- [4] M. Yoshimura, H. Kawai, T. Iyota and Y. Choi: *FPGA-Based Image Processor for Sensor Nodes in a Sensor Network*; The Open Signal Processing Journal, 1876-8253, Vol(7),2020, (2009)
- [5] 小橋正, 伊与田健敏: 自律光学航法で使用される参照画像生成の高速化を目的とした Open GL ライブラリの解析, 創価大学大学院修士学位論文 (2021)
- [6] D. Patterson and A. Waterman: *The RISC-V Reader An Open Architecture Atlas*, Strawberry Canyon (2017)
- [7] <https://github.com/riscv/riscv-gnu-toolchain.git>

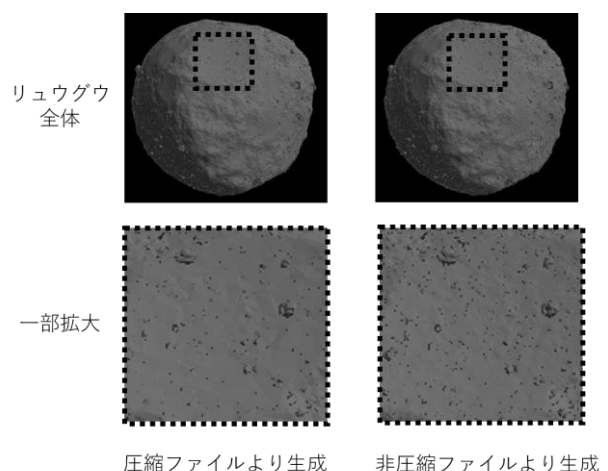


図 2 生成した参照画像の比較