

要 旨

超解像処理の Haskell による実装と並列化に関する研究

松本 拓也

関数型言語 Haskell は遅延評価による評価戦略や参照透過性の保証などの特徴を持ったプログラミング言語である。Haskell によるプログラミングでは、手続き型プログラミングと比べて簡潔な記述ができるようになることや、並列化が容易であることなどのメリットがある。Haskell は近年ではライブラリやドキュメント、コミュニティなどが発達し並列処理を含むアプリケーション開発など様々な用途・環境で利用できるようになってきた。

本研究では、fMRI 画像を対象とした超解像処理を Haskell により実装し、Haskell によるアプリケーション開発及び並列処理による高速化とその効果について調査した。本研究では逐次処理、Repa ライブラリによる CPU を用いた並列処理、Accelerate ライブラリによる GPU を用いた並列処理の実装し、比較実験を行った。

実験の結果、CPU による並列処理では先行研究で用いられた Java と今回実装した Haskell による逐次処理に対し、処理時間が最大で約 7 割短縮された。一方で、Accelerate による GPU 並列処理実装では、処理時間約 2 割した。

GPU による並列処理ではあまり効果が得られなかった原因として SIMD の発散が起きた可能性や CUDA カーネルのキャッシュが働かなかったなどの可能性がある。

以上の結果から、Haskell を用いた並列実装において、Repa を用いた場合には逐次処理のプログラムから大きな改変を行わずに実装ができ、並列化の効果も十分に出すことができることがわかった。一方 Accelerate を用いた場合、実装に関しては Repa と実装コストに大きな差はないが、逐次処理のアルゴリズムを流用するだけでは遅くなってしまいう問題が発生する。

キーワード Haskell, 並列処理, 超解像処理

Abstract

A Study on the Implementation and Parallelization by Super-resolution Processing Using Haskell

Takuya MATSUMOTO

Functional programming language Haskell has several features such as lazy evaluation and referential transparency. Advantages of Programming in Haskell, include brief description compared with procedural programming and parallelization. In recent years, thanks to the development of libraries, documents and community, we can develop a variety of application, including parallel processing.

In this study, I implemented the super-resolution processing for the fMRI images using the Haskell. Then I verified the performance and the effect of parallel processing, I implemented super-resolution processing using normal Haskell (sequential), Repa and Accelerate libraries (oarakkek), and compared them.

As a result of experiments, parallel computation with a multicore CPU reduced 70% of the processing time, parallel computation with GPU, however increased the processing time by 20%. This is probably caused by missing the cache of CUDA kernels or by the SIMD divergence.

The results give the following findings. In the parallel implementation using the Haskell, it is easy to implement sequential program ones. Furthermore, those parallel programs exhibit nice scalability. On the other hand, there remains an issue that only using the sequential processing algorithms dose not benefit from the high performance of GPUs.

key words Haskell, Parallel processing, Super-resolution