

要旨

超解像処理を題材とした プログラム高速化の効果の検証

畑中 涼

近年の計算機はキャッシュやパイプライン処理などにより高速化を実現している。そのため、計算を高速化するには、そのような機能を有効活用する必要がある。

本研究では、超解像化を行う画像処理プログラムを題材にし、最適化手法をいくつか適用してプログラム高速化を行う。まず、元のプログラムである Java プログラムを C++プログラムに書き換えた。その後、書き換えた C++プログラムに対して各最適化手法を施すことで高速化を狙った。評価実験では、元の Java プログラムとそのまま書き換えた C++プログラム、最適化を施した高速 C++プログラムの 3 種のプログラムの処理時間を計測し、比較した。この結果、高速 C++プログラムの処理時間は Java プログラムの処理時間の 27%にまで短縮された。また、各最適化手法別で処理時間を計測し、処理時間と書換えコストの観点から比較した。その結果、題材にしたプログラムでは、コンパイラに定数伝播を促す書き換えが最も効率的な最適化手法であることを確認した。

以上の結果から、最適化を行うことで処理時間は短くなり、高速なプログラムにすることが可能であることが確認できた。また、書換えコストを考慮した場合、コンパイラに定数伝播を促す書き換えを行うことで、最も効率的にプログラムの高速化を実現できることがわかった。

キーワード プログラムの書換え, 高速化, 最適化, 画像処理

Abstract

Verification of Effectiveness of Program Optimization Using Super-resolution Processing as a Case Study

Hatanaka Ryo

Recent computers realizes high performance with the cache and pipelining mechanisms. Therefore, in order to perform high-performance computing, it is necessary to use such mechanisms effectively.

In this study, I tried to accelerate super-resolution processing program by applying program optimization. First, I translated the Java program into a C++ program. Then, I tried to accelerate the translated C++ program by four program-rewriting techniques. I measured the processing time of the existing Java program, the translated C++ program, and the accelerated program. As a result, processing time of the accelerated C++ program reduced to 27% of that of the existing Java program. Furthermore, I measured and compared processing time and rewriting cost for each optimization technique. As a result, I found that to promote the compiler to do the constant propagation is the most efficient optimization.

From these results, We can shorten a processing time by performing program optimizations. In particular to promote the compiler to do the constant propagation is the most efficient optimization with respect to the rewriting cost.

key words Rewriting programs, Acceleration, Optimization, Image processing