

GPUコンピューティングを用いた 組み込み画像処理高速化技術

Development of High-Speed Technology for Embedded Image Processing Using GPU Computing

坂 匡 晃* 上 田 滋 之* 中 村 哲 平*
Masaaki SAKA Shigeyuki UEDA Teppei NAKAMURA

要旨

近年、3Dグラフィックスなどの描画処理専用であったGPU (Graphics Processing Unit) をグラフィックス処理以外の汎用計算に利用する技術としてGPUコンピューティング (またはGPGPU [General Purpose GPU]) が注目を集めており各分野で活用されている。

コニカミノルタの複合機は高機能、高性能、高画質を望むユーザーニーズに対応してきた結果、CPU負荷の増大が課題となっている。複合機の数ある機能の中から今回我々が着目したのは、コンパクト (高圧縮) PDFフォーマットのスキャン配信機能である。コンパクトPDFとは文字領域と写真領域を高精度で分離し、それぞれに異なる圧縮方法を適用することで、原稿の文字可読性を維持しつつ高圧縮なPDFファイルが生成可能な技術であり、画像処理部分はコニカミノルタ独自のアルゴリズムである。しかし、処理が複雑なためJPEGやPDFフォーマットに対してパフォーマンス面で不利となっていた。

そこで我々は、GPUとCPUといった異なる種類のコアが混在するヘテロジニアス環境にて動作可能な並列化フレームワークの一つであるOpenCL C言語を用いてコンパクトPDFの画像処理を実装し、複合機に搭載されている組み込み向けGPUを使った高速化を目指した。

複合機に搭載されるGPUコア数の制限がある中、

- ・ OpenCLを用いた並列化向けコードへの最適化
- ・ CPU/GPUメモリ共有化によるメモリ使用の効率化

の2つの技術を開発した。

その結果、コンパクトPDFスキャンにおいて、従来CPU比で大幅なパフォーマンス向上とCPUの負荷低減を達成し、ユーザーのスキャン待ち時間低減に貢献できた。

また本開発の成果として、GPUを複合機の組み込み画像処理に適用することの有用性を確認でき、ベースシステムの構築を完了させることができた。今後、複合機や複合機と連携したソリューションにおいて、新たな顧客価値を提供する手段としてGPUコンピューティングは様々な機能に応用展開できると考えている。

Abstract

GPU's (graphics processing units), originally used only for drawing processing, are now also tapped in support of general-purpose calculating. Because the superior functionality, performance, and quality of Konica Minolta MFP's, the load on their CPU's has significantly increased and invited the use of general-purpose GPU's.

A prime example of this use is in accommodating the compactPDF format. The compactPDF format accurately separates a document's text regions from its graphic regions and applies specific compression methods to each, producing high-compression PDF files that yet maintain document readability. However, the complexity of compactPDF processing inhibits performance, which lags behind that of such formats as JPEG and PDF. To compensate, we turned to general-purpose GPU's.

Our image processing of the compactPDF format's scan-and-deliver function is based on a unique Konica Minolta algorithm. Utilizing an embedded GPU, we employed OpenCL C language, which provides a parallel framework operable within a heterogeneous environment, a framework in which differing cores (CPU's and GPU's) are teamed. The number of GPU cores that can be embedded in an MFP is limited, but two technologies overcame this: 1) OpenCL C language to optimize parallel code, and 2) CPU/GPU memory sharing to maximize memory use efficiency. This so reduced the CPU load that performance over solo CPU processing improved exceptionally, as did user wait time. With these benefits confirmed, we then constructed a base system.

Our experience indicates that CPU/GPU integration is a productive path in supporting the various simultaneous functions of an MFP and achieving the ultimate objective: value for the customer.

* 開発本部 第1 OPシステム制御開発部

1 はじめに

コニカミノルタのオフィス向け複合機は、様々なユーザーニーズに対応すべく複合機の高機能、高性能、高画質化を追求してきた。また最近では、モバイル端末やクラウドサーバーの普及に伴い、これらを複合機と連携させたソリューションサービスのニーズが増え、複合機が担う役割はますます大きくなっている。

複合機のコピー、スキャン、ファックスなどの機能においては、搭載されたCPUにて様々な画像処理が実施されているが、複合機はプリント中のスキャンといったマルチ同時が一般的なため、こういったケースではCPUのリソース不足が顕著になり、負荷低減が求められていた。

こうした中、近年3Dグラフィックスなどの描画処理専用であったGPU (Graphics Processing Unit) をグラフィックス処理以外の汎用計算に利用する技術、GPU コンピューティング (またはGPGPU [General Purpose GPU])、が注目を集めており、様々な分野で活用されている。そこで我々は、CPUに代わる新たな画像処理手段として、複合機の画像処理に対してOpenCLを用いたGPU コンピューティング技術の適用を検討し、CPUの負荷分散と、画像処理のパフォーマンス向上を目指した。

2 GPU適用機能の選定と従来の課題

GPU コンピューティングの適用機能として、我々はスキャン機能のコンパクトPDFフォーマット生成処理を選定した。コンパクトPDFとは、原稿の文字可読性を維持しつつ高圧縮なPDFファイルが生成可能な技術であり、画像処理部分はコニカミノルタ独自のアルゴリズムとなっている。処理の概要は、原稿をスキャンして得られたRGBデータと、hardware (以下HWと略す) によって抽出した画像の領域を識別する属性データを用いて、文字領域と写真領域を高精度で分離し、それぞれに異なる圧縮方法を適用することでPDFファイルを生成するものである。

しかしコンパクトPDFのデメリットとして、処理が複雑で演算量が多く、Fig. 1 に示すようにパフォーマンス面でJPEG、PDFといったフォーマットに対して劣るという課題があった。

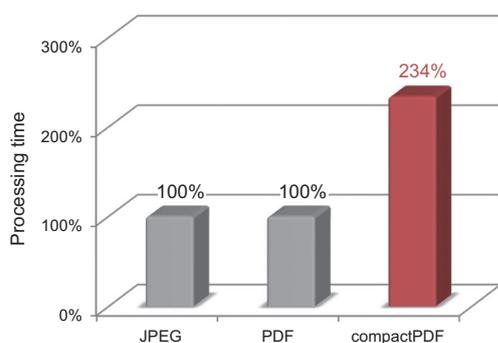


Fig. 1 CPU scanning performance. With CPU scanning alone, compact PDF processing takes far longer than PDF or JPEG processing.

次にFig. 2にコンパクトPDF生成の処理フローを示す。今回我々は、演算負荷が高いという観点から、GPUを適用する処理として「文字／写真領域分離処理」と「圧縮処理」を選定した。

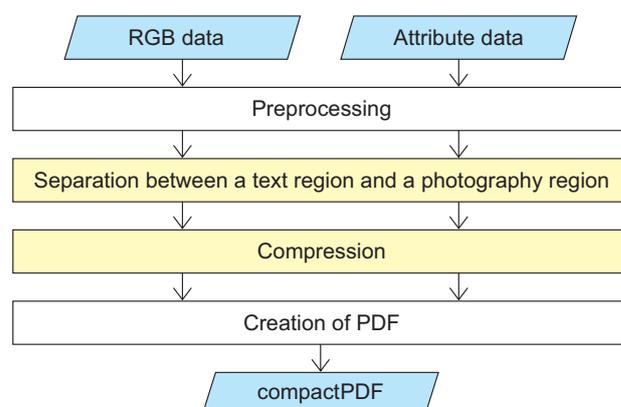


Fig. 2 A process of the compactPDF.

「文字／写真領域分離処理」は、対象画素を中心に周辺画像を広いエリアを使って処理するアルゴリズムとなっているため、HW処理には不向きであり、software (以下SWと略す) 処理による高速化が求められていた。

SW処理による高速化の手段として、一つはCPUの性能アップがある。しかしながら近年、CPUコア性能は頭打ち状態で、大幅な動作周波数アップも見込めないことから、CPU単体性能に頼った高速化が難しくなっている。

もう一つの手段はCPUのマルチコア化であるが、マルチコアでは各コア間の同期に伴うオーバーヘッドが大きく、またコアあたりの同時処理できるスレッド数が少なくなっている。一方GPUはコア毎に数百以上ものスレッドを同時に処理できるため画像処理のような画素毎の並列動作には優れている。

3 複合機へのGPU適用の課題

複合機のような組み込み機器では、搭載できるGPUコア数に制限があるため、その中で如何に効率良く並列演算できるかが重要であった。またGPUを使用することでCPUに比べてワークメモリ増加が想定されるが、複合機のような組み込み機器は厳密で最小限のメモリ割り当てとなっているため、メモリ増加を極力抑えた上で複合機のメモリ管理構造に合った設計が必要であった。

これらの課題から、複合機へのGPU適用に向けて技術開発すべき内容を下記2点とした。

- 1) 並列演算性能を最大限発揮するためのコード最適化と処理アルゴリズムの見直し
- 2) CPU/GPUメモリ共有化によるメモリ使用の効率化

以降では、上記2点に対する技術開発の内容を紹介していく。

4 開発技術

4.1 並列化手法の紹介

GPU コンピューティングのための動作手法として、OpenCL (Open Computing Language) を使用した。OpenCLとはCPU+GPUといったヘテロジニアスな環境に適した並列プログラミングのためのフレームワークであり、KhronosGroupによって仕様の標準化が行われている。

OpenCLは、プロセッサの種類やベンダーに依存することなく、統一された並列ソフトウェアを実現可能であり、Fig. 3 に示すようにホスト側 (CPU) からOpenCL ランタイムAPIによりデバイス側 (GPU) を制御する。デバイス側 (GPU) で動作するプログラムをカーネルプログラムと呼び、OpenCL C 言語にて記述される。



Fig. 3 Illustration of OpenCL (open computing language).

4.2 並列化向けコードへの最適化

元々のCPU用ソースコードは並列化を考慮したコードとなっていなかったため、GPUにて効率的な並列動作を実現するため以下3つの方策を実施した。

4.2.1 ベクタ型を用いたSIMD演算

GPUは128bit幅の内部レジスタを有しており、本レジスタを使用し、単一命令で複数データに対して処理するSIMD (Single Instruction Multiple Data) 演算器が搭載されている。

OpenCL C 言語では同じ型の複数のデータを同時に扱うことができるベクタ型が定義されており、ベクタ型を使用することでGPUでのSIMD演算が可能となる。

組み込み画像処理で使用される画素データは1画素8bit (0-255階調) で扱われる場合が多く、コンパクトPDFも同様のため上記SIMD演算器を使用することで、CPUで行うスカラー演算に対して、命令数を最大で1/16に削減し、高速化を達成した (Fig. 4)。

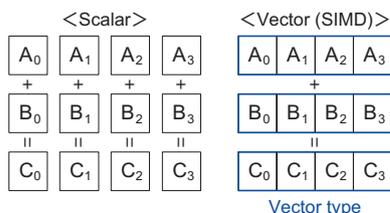


Fig. 4 SIMD calculation using a vector type.

4.2.2 並列化可能なアルゴリズムへの改編

コンパクトPDFの文字写真分離処理では、属性情報を参照しながら、RGBの各画素の処理内容を切り替えるため、画素毎に条件分岐処理を行うこととなる。

一方で、GPUでは前記ベクタ型を使用したSIMD演算による複数の画素への同一処理が想定されているため、条件分岐を伴うような複雑なアルゴリズムではパフォーマンスが出にくいという特徴があった。

そこで、画素毎に異なる処理を行うアルゴリズムにおいては、条件分岐の真偽両方の結果をSIMD演算にて算出し、その結果に対して論理演算 (AND/OR) によるマスク処理によって取捨選択するようなアルゴリズムに変更した。

これにより、各画素毎行っていた条件分岐処理を削減し、条件分岐処理の含まれる複雑な処理に対してもSIMD演算が可能となった (Fig. 5)。

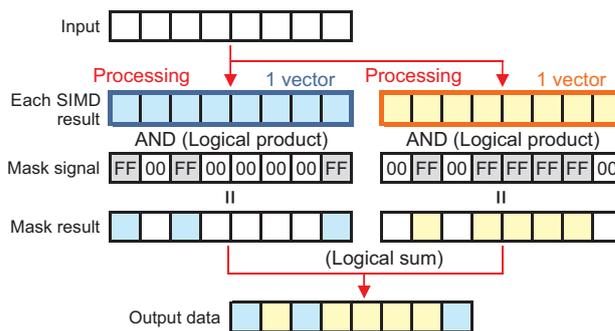


Fig. 5 Change into a parallelizable algorithm.

4.2.3 CPU/GPU 並列動作可能なアーキテクチャ再構築

文字/写真領域分離処理では、スキャンRGB画像とは別に文字色毎に分離した属性情報も使用している。一連の処理の中には、属性情報の文字色毎に独立して演算可能なものも含まれるため、これらを抽出し、最も効率良く並列動作可能となるようアーキテクチャを再構築した。

Fig. 6 に示す例では、RGB画像と各色属性に対する処理を処理の特性からCPUとGPUを最適に分担させることで並列動作を実現している。

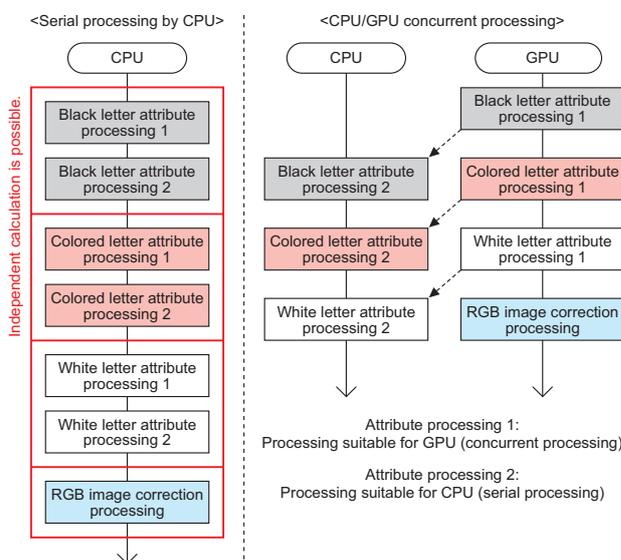


Fig. 6 Illustration of CPU/GPU concurrent motion.

4.3 CPU/GPUでのメモリ共有化：複合機のメモリ管理構造

カラー複合機のスキャン処理は、A3サイズ、600dpi、1ページあたり約200MBと膨大な画像データを扱うため効率的なデータ転送と画像圧縮が必要となる。そのためメモリに関しては連続領域を静的に確保し、1ページをバンドと呼ばれる小領域に分割することで、高速かつ低メモリ使用量を実現しており、GPUを適用する場合にもこうしたメモリ管理構造に対応することが前提条件となる。Fig. 7 にスキャン処理で作成したバンド画像を複合機のファイルメモリに転送するまでの処理フローを示す。

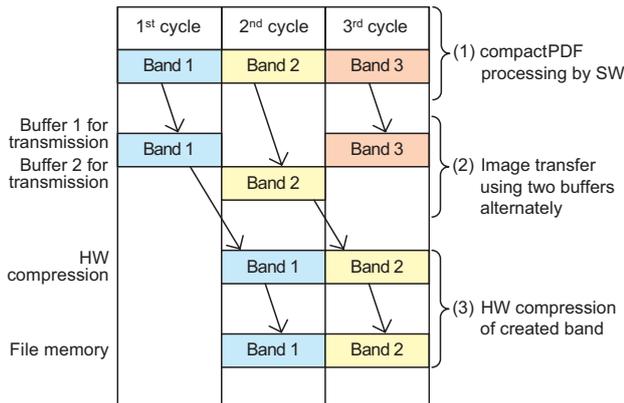


Fig. 7 Processing flow of image data (a band method).

しかしながらGPUの制御用APIであるOpenCLでは標準で連続領域を確保することが出来ないという課題があった。標準のGPUドライバを使用した場合、使用メモリはヒープ領域から確保されるため、これではCPU用のバンド領域と併せて余分なメモリ領域が必要となり、更に両メモリ間でコピー処理が必要となるため、パフォーマンス低下にもつながっていた。そこで我々は、バンド領域と同一アドレスをGPUメモリ領域としてマッピング可能となるようGPUドライバをカスタマイズし、画像処理の高速化と消費メモリサイズの削減を両立した(Fig. 8)。

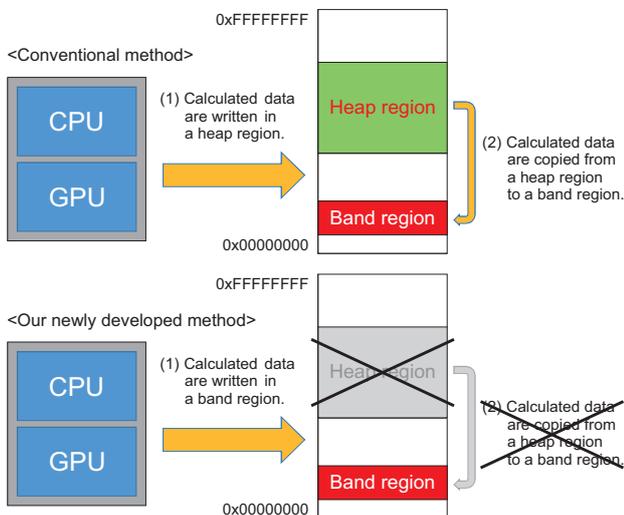


Fig. 8 Change of a memory securing method.

5 検証結果

5.1 パフォーマンス検証結果

GPUを適用した「文字／写真領域分離処理」と「圧縮処理」のパフォーマンス検証結果をFig. 9に示す。CPU比-69.0%という大幅な高速化を達成した。

使用した検証環境は1コア400MHzのGPUであるが、GPU向けのコード最適化により、複数のスレッドを効率良く並列化処理させることでコアの性能を最大限引き出した結果と言える。

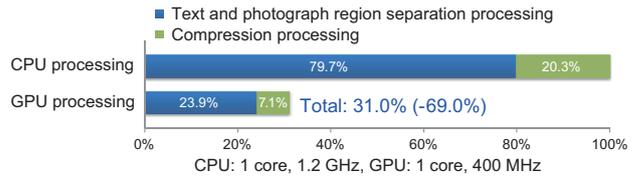


Fig. 9 Results of performance verification using GPU.

次にFig. 10にコンパクトPDFスキャンにおける、複合機のスタートキー押下から、ファイル送信完了までの処理時間を示す。CPU処理ではJPEG、PDFと比べて+45%と大きくパフォーマンスで劣っていたが、GPU化によりJPEG、PDFと同等のパフォーマンスまで高速化ができており、ユニカミノルタのコンパクトPDFスキャンにて、低ファイル容量とパフォーマンスの両立が可能となった。

さらにFig. 11に示すように、コンパクトPDFスキャン時のCPUの負荷低減にも大きく貢献でき、CPU側で別タスクが実行可能となったことから、マルチ機能の同時実行時のパフォーマンス向上にも繋がった。

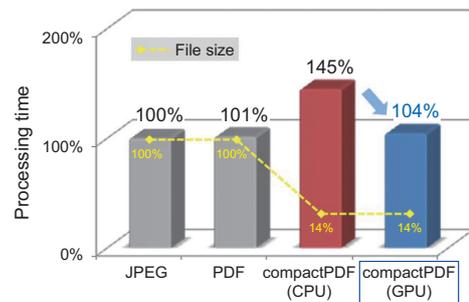


Fig. 10 Comparison of scanning performance.

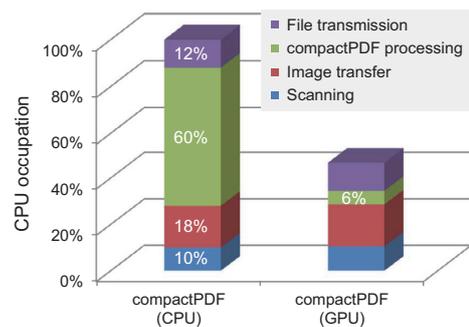


Fig. 11 Comparison of CPU occupation percentage. The CPU occupation at compactPDF processing is greatly decreased.

5.2 メモリ使用量検証結果

Fig. 12 にCPUとGPUのメモリ使用量の比較を示す。通常GPUが使用するメモリはヒープ領域から確保されるため、システムのメモリ使用量がCPU比+111%まで増加していた。

本開発にてバンド領域でCPUとGPUがメモリ共有可能な仕組みを開発することで、GPU適用時のメモリ増加量をCPU比+17%まで抑制することができ、システム全体のメモリ使用量増加を防ぐことができた。

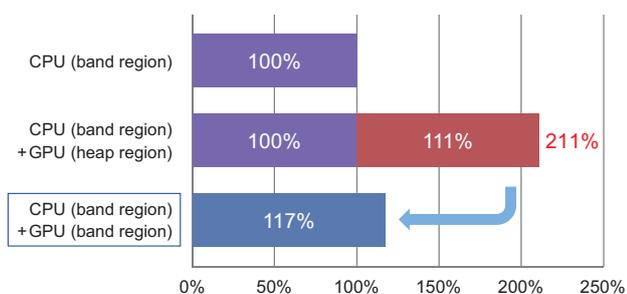


Fig. 12 Result of verification of memory use volume.

6 まとめ

本技術開発にて、GPUを複合機の組み込み画像処理に適用する際のパフォーマンスとメモリ使用量の技術課題をクリアし、CPU負荷低減とパフォーマンスの大幅向上を達成できた。今回の結果から複合機の画像処理にGPUを適用することの有用性を実証することができ、ベースシステムの構築を完了させることができた。

今後も半導体プロセスの微細化が進み、複合機やモバイル端末といった組み込み機器向けに搭載可能なGPU性能は向上していくと予想される。

我々は、複合機や複合機と連携したソリューションにおいて、拡大していく顧客ニーズに対応する新たな価値提供手段として、今後もGPUコンピューティングを様々な機能に応用展開していく。