ISS-SP-333

GPGPU を利用した高速透視化マルチウィンドウシステムの X Window System への組み込み

森　洋史†　　有本　和民††　　横川　智教††　　佐藤　洋一郎††
† 岡山県立大学大学院情報系工学研究科　　†† 岡山県立大学情報工学部

1. まえがき

本研究では，GPUを汎用的に利用するGeneral Purpose computing on GPU(GPGPU)を利用した高速透視化マルチウィンドウシステムをX Window System(X)に組み込みを行った[1].

2. 透視化マルチウィンドウ合成原理

透視化マルチウィンドウの合成原理を図 1 に示す．操作対象となるウィンドウ(TW)，TW 以外のウィンドウ(NTW)の合成ウィンドウ(NTMW)，及び背景ウィンドウ(BG)の画像情報をそれぞれ画像情報メモリ(BM)，前者 2 つの領域情報をそれぞれ領域情報メモリ(FM)によって管理する．

まず，BM_{NTMW}，BM_{BG}，及び FM_{NTMW} を並列に読み出し，加重加算演算を施すことで合成画像を生成する．そして，この画像とBM_{TW}，及びFM_{TW}に対して，同様に加重加算演算を施すことで，マルチウィンドウ画像を生成する．

3. CUDA を利用したマルチウィンドウ合成

上述したマルチウィンドウ合成処理を GPGPU の 1 つのアーキテクチャである Compute Unified Device Architecture(CUDA)で実現する．また，合成画像のフレームバッファへの転送は，OpenGL の機能の1つである Pixel Buffer Object(PBO)で実現する．そして，図 2 に示すように，CUDA 空間への 5 つのメモリのデータ転送は，Mapped Memory によるデバイスメモリ空間とホストメモリ空間の共有によって実現する．

4. X Window System の管理サーバの改造

X では，各スクリーンの情報は ScreenRec 構造体によって各スクリーン毎に管理し，ScreenInfo 構造体によってまとめて管理している．これに対し，本システムでは，上述したように，2 段階に分けてマルチウィンドウ合成を行うため，BM_{TW}，BM_{NTMW}，BM_{BG}，FM_{TW}，及びFM_{NTMW}をそれぞれ異なる ScreenRec 構造体で管理する．そして，TW，NTW 及び BG のそれぞれの WindowRec 構造体に対し，対応する ScreenRec 構造体を関連付ける．その後，マルチウィンドウ合成を行う．

5. 検討

提案法におけるウィンドウ操作の実行時間と既存システムである X におけるそれとの比較を行った．CentOS6.0，Intel Core i3 とNVIDIA GeForce GT430 を搭載した PC で測定を行った結果を表 1 に示す．この結果によれば，透視化処理を含む移動及びスクロールでは約 1/10 の，リサイズでは約 1/8 の処理時間の短縮が期待できる．

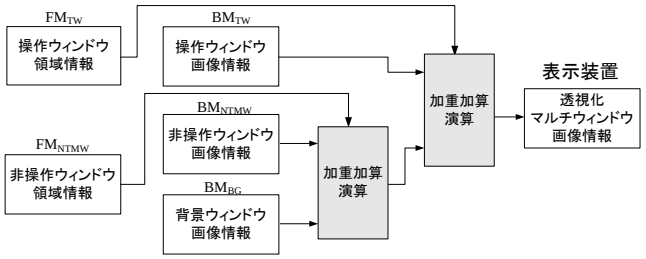

図1. 透視化マルチウィンドウの合成原理

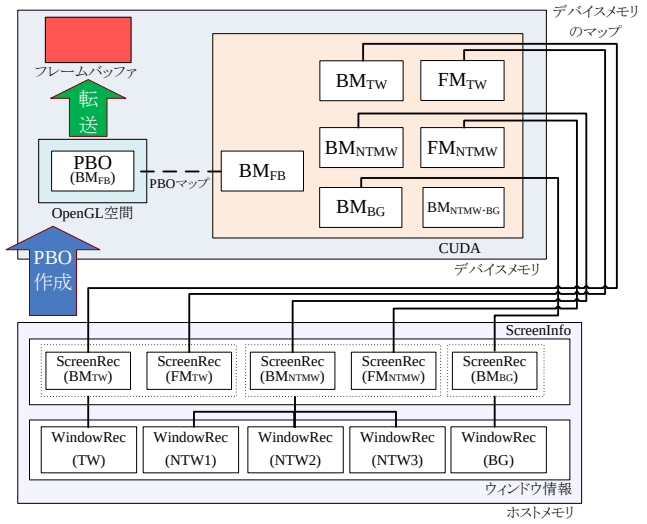

図 2. マルチウィンドウ合成原理の概要

表 1.ウィンドウ操作実行時間(単位:msec)

本システム			X Window System
移動	スクロール	リサイズ	
1.5	1.4	1.9	16

6. あとがき

GPGPU を利用した透視化マルチウィンドウシステムを X Window System に組み込む方法について検討した．残された課題は，動画ウィンドウの表示等への対応が挙げられる．

参考文献

[1] 森洋史:”X Window システムにおける透視化マルチウィンドウ合成の GPGPU を利用した高速化”，平成 25 年度 岡山県立大学大学院情報系工学研究科 修士論文(2014-02)