

C-024

ウィンドウの透視化と輝度低下機能を持つマルチウィンドウシステムの評価 Evaluation of a multi-window system with Ability of Translucentizing and lowering the brightness of Windows.

宮崎 仁[†] 茅野 功[†] 佐藤 洋一郎[‡] 横川 智教[‡] 早瀬 道芳[‡]
Hisashi Miyazaki Isao Kayano Youichiro Sato Tomonori Yokogawa Michiyoshi Hayase

1. まえがき

筆者らはこれまでに、操作の対象となるウィンドウの透視化機能を持つマルチウィンドウの合成方法について検討を行ってきた [1]。この合成方式によると、ホストへの負荷をほとんどかけることなくウィンドウの透視化や輝度の低下を実現できるが、その処理時間についての検討はまだ行っていない。そこで本稿では、このマルチウィンドウ合成原理に基づいたマルチウィンドウシステムのプロトタイプを用いて、ウィンドウの透視化画像あるいは輝度低下画像の生成・表示に必要な処理時間を測定し、従来のマルチウィンドウ合成方式におけるソフトウェア的手法と比較・検討をする。

2. 透視化機能を持つマルチウィンドウシステム

本稿のマルチウィンドウ合成原理を図1に示す [1]。W₁が操作の対象ウィンドウ(以下、TW)であり、W₂とW₃が操作の対象外ウィンドウ(以下、NTMW)を構成する。

- BM_{TW} : TWの完全な画像を格納するメモリ
 BM_{NTMW} : NTMWを合成した画像を格納するメモリ
 BM_{BG} : 背景の完全画像を格納するメモリ
 FM_{TW} : TWの完全な領域を、領域内部であれば対応するウィンドウ番号、領域外であれば0(背景ウィンドウの番号)として格納するためのメモリ
 FM_{NTMW} : NTMWを構成する各ウィンドウと背景ウィンドウの領域上にFM_{TW}と同様の形式で各ウィンドウ番号を格納するためのメモリ
 OPC-A,B : NTMW及びTWの画像に対して透視化を行うための演算回路
 FS : 画像合成を行うための2-1マルチプレクサを主な構成要素とする画像選択回路
 REG-A,B : TW・NTMW双方、NTMWのみ、及びTWのみを透視化する場合、それぞれ、00, 01, 及び10に設定する。また、通常表示、及び輝度低下の場合には11に設定する。

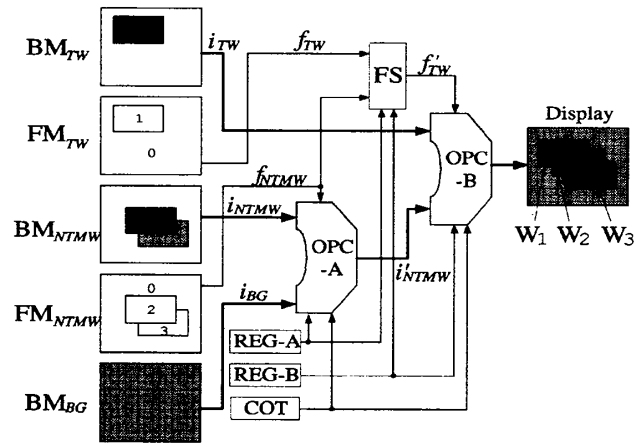


図1: マルチウィンドウ合成原理

- COT : 透視化及び輝度低下を行う際の重みである乗率 (0より大きく1以下) をMSBが整数部、残りが小数部という形式で格納するためのメモリ (通常表示の場合全て1に設定)

BMはいずれも表示装置1画面分に対応するアドレス空間を持ち、その深さは画像の色の深さと同じである。以下、FMに格納される情報のことを領域情報と呼ぶ。

表示装置のフレーム周期毎に、BM_{NTMW}上の画像及びBM_{BG}上の画像は、0番地(画面左上に対応する)から表示装置の走査に同期して、それぞれ、シリアル信号*i*_{NTMW}及び*i*_{BG}として読み出される。これと同様に、FM_{NTMW}の領域情報も0番地からシリアル信号*f*_{NTMW}として読み出される。COTの値を α とすると、OPC-Aは各画素毎に*i*_{NTMW}と*i*_{BG}に対して α , $1 - \alpha$ を乗算し、さらにそれらを加算することにより透視化のための演算を施し、また輝度低下の場合は、*i*_{NTMW}に α を乗算したその結果をシリアル信号*i'*_{NTMW}として出力する。これと同期して、BM_{TW}上の画像及びFM_{TW}上の領域情報も、それぞれ、シリアル信号*i*_{TW}及び*f*_{TW}として読み出される。ただし、これらの読み出し開始アドレスは、ウィンドウ操作にともなう表示画面上でのウィンドウ位置に応じて0番地から移動する。FSはREG-A=1, REG-B=0, *f*_{NTMW}=0の時のみ0を、これ以外の場合には*f*_{TW}を*f'*_{TW}として出力する。これを制御信号としてOPC-BによりOPC-Aと同様の演算を行いマルチウィンドウを合成し表示装置に出力をする。

ウィンドウの透視化処理、あるいは輝度低下処理に関

[†]川崎医療短期大学 臨床工学科
[‡]岡山県立大学 情報工学部

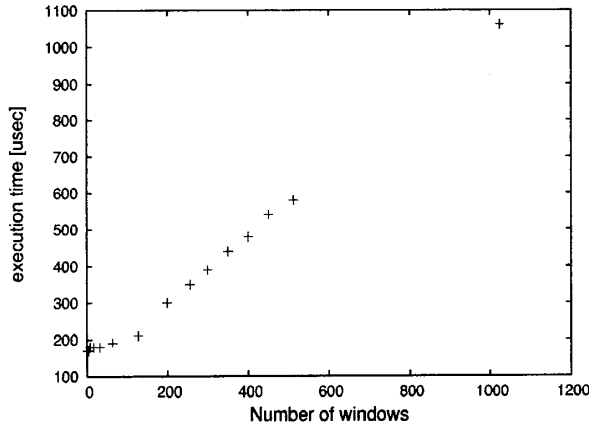


図 2: 本方式における透視化及び輝度低下処理時間

わる画像演算処理はすべてハードウェアで行われ、ソフトウェアは COT 及び REG-A, B への書き込み処理だけで実現できるため、これらの画像変化、およびその解除が高速に処理可能となる。

3. 処理時間に対する検討

3.1 本方式による処理時間

前述のマルチウィンドウシステムを Microsoft 社 WindowsNT4.0 を搭載したパーソナルコンピュータ（主記憶：96Mbyte, CPU：Intel Pentium Pro 200MHz）に実装し、ウィンドウの透視化処理及び輝度低下処理の所要時間を測定した。背景ウィンドウの大きさは 512 画素 × 480 画素であり、処理の対象となるウィンドウの大きさは任意とし、表示ウィンドウ数は 1~1024 とした。

処理時間の測定結果を図 2 に示す。処理時間は、ウィンドウ数に比例して増加している。しかし、ソフトウェアが行う処理は、透視化、及び輝度の低下いずれの場合においても COT, REG-A, B への書き込みのみであるため、対象となるウィンドウの大きさは処理時間に影響を与えず、表示ウィンドウ数が 1000 の場合でも約 1000[μsec]であった。但し COT 及び REG-A, B の更新は垂直帰線期間に行われるため、表示期間に突入した直後にこれらの書き換え命令が起こると、1 フレームの表示期間だけ待機することになる。したがって、システムが処理を完了し画面に表示するまでの最大所要時間は 1 フレーム周期、つまり 10.6msec に相当する。

3.2 描画時合成方式による透視化処理時間

本方式との比較として既存の合成方式（描画時合成方式）における透視化の所要時間の測定を行った。描画時合成方式を採用している代表的な OS である Microsoft 社 Windows2000 上で任意のウィンドウ画像を透視化するアプリケーションを作成し、実行時間を測定した。このとき使用するパーソナルコンピュータに実装するメモリを 2 種、及び CPU を 4 種それぞれの組み合わせについて測定を行った。

その結果を図 3 に示す。処理時間はメモリの量にほとんど依存せず、CPU のクロック数による影響が大きいことがわかる。また処理時間は、対象となる画素の大き

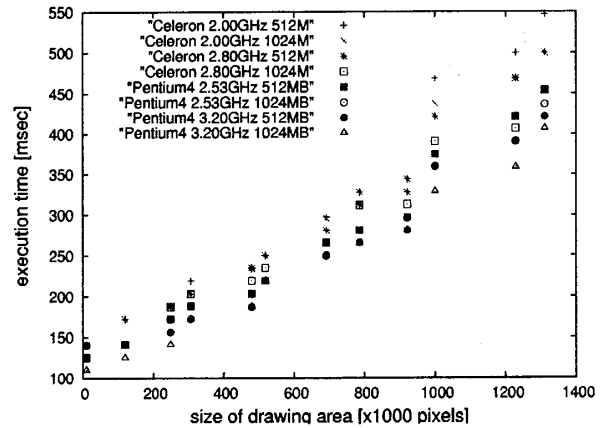


図 3: 描画時合成方式における透視化処理時間

さに比例して増加している。

同様に、描画時合成方式における輝度の低下についても透視化の場合と同じ条件において測定を行った。その結果、所要時間は透視化の場合の約 0.7 倍であった。これは、透視化の場合その対象となる画像領域の重なり合う 2 つの画像をそれぞれ演算し、さらにその合成を行う必要があることに対して、輝度低下の場合その対象となる画像のみの演算で済むためと考えられる。透視化の場合と同様にその処理時間は、メモリの量にほとんど依存せず、CPU のクロック周波数に依存し、また、対象となる画素の大きさに比例して増加することがわかった。

3.3 考察

上述の通り、描画時合成方式においてウィンドウの透視化あるいは輝度低下処理を行う場合、その対象となる画素数が処理時間を決定する主要因となることに対し、本方式では、処理の対象となるウィンドウの数が主要因となることがわかる。また、512 × 480 画素を対象とした透視化処理の場合、Pentium4 3.2GHz の CPU を搭載するコンピュータを使用しても、その処理時間は約 140[msec] を要することから、本方式がこれらの処理に劇的な高速化をもたらすことがわかる。さらに本方式では、画素数が増加しても CPU のクロック周波数に依存せず、常に一定の時間で処理可能であり、かつ高速に処理を行えるため、動画像の透視化及び輝度の低下を行う場合においても非常に有用である。

4. あとがき

本稿では、ウィンドウの透視化と輝度低下機能を持つマルチウィンドウシステムの評価を行った。今後、このシステムを Linux ベースのパーソナルコンピュータに実装し、ハードウェアの改良を行い、より高解像度表示を可能にするとともに、さらなる評価を行う予定である。

参考文献

- [1] 茅野功, 田辺勝也, 佐藤洋一郎, 横川智教, 早瀬道芳: "操作対象ウィンドウの透視化機能を持つマルチウィンドウシステム", FIT2004 論文集, 第一分冊, pp.265-266, Sept. 2004.