C-019

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
A Multi-Window System with Ability of Translucentizing the Windows

茅野　功† 田辺　勝也‡ 佐藤　洋一郎‡ 横川　智教‡ 早瀬　道芳‡
Isao Kayano Masaya Tanabe Youichiro Sato Tomonori Yokogawa Michiyoshi Hayase

1. まえがき

本稿では，操作対象ウィンドウに隠されたウィンドウを表示できる，すなわち，透視化機能を持つマルチウィンドウシステムの実現法を提案している．本システムでは，高速なラスター演算の実現が容易であるという点に着目して文献 [1] の方式を改良し，操作対象ウィンドウの画像を格納するメモリ，それ以外のウィンドウを描画時合成方式に基づいて合成した画像を格納するメモリ及び，背景画像を格納するメモリから並列かつ走査に同期して読み出される３つの画像を用いて表示時合成方式の原理に基づいてマルチウィンドウ画像を合成する．透視化機能は，３つのメモリの出力側に適宜画像演算を行なう回路を付加することにより，ホストへの負荷をほとんど与えることなく高速に実現している．

2. マルチウィンドウ合成原理

マルチウィンドウ合成原理を図１に示す．W_1 が操作対象ウィンドウ（以下，TW）であり，W_2 と W_3 が操作対象外ウィンドウ（以下，NTMW）を構成する．BM_{TW}，BM_{NTMW} 及び BM_{BG} は，それぞれ，TW の完全な画像，NTMW 画像及び背景ウィンドウの完全な画像を格納するためのメモリである．これらのメモリ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．また，FM_{TW} は，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である．FM_{NTMW} は，NTMW を構成する各ウィンドウと背景ウィンドウの仮想可視領域を FM_{TW} と同様の形式で格納するためのメモリである．以下，FM_{TW} と FM_{NTMW} に格納される情報のことを領域情報と呼ぶ．OPC-A 及び OPC-B は，それぞれ，NTMW 及び TW の画像に対して透視化を行うための演算回路であり，FS は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である．また REG-A，REG-B 及び COT は画像の透視化等の画像演算処理を行う際に参照されるレジスタ及びメモリである（**3.** で詳述）．これらは本来，OPC-A 及び OPC-B 内部の構成要素であるが，便宜上図１にも記している．

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．OPC-A は各画素毎に i_{NTMW} と i_{BG} に対して透視化のための演算を施し（**3.** で詳述），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

†川崎医療短期大学　臨床工学科
‡岡山県立大学　情報工学部

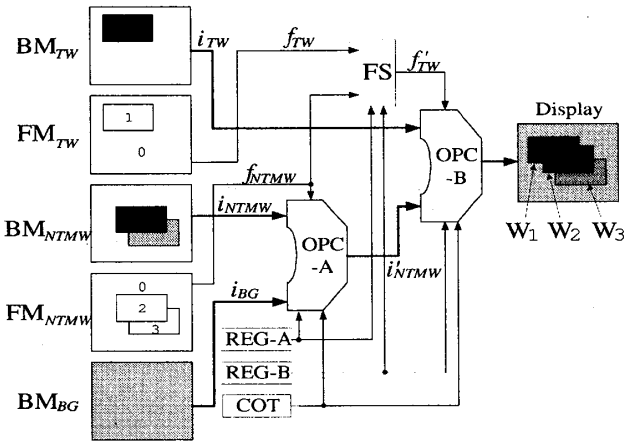

図 1: マルチウィンドウ合成原理

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．２つの領域情報 f_{TW} 及び f_{NTMW} は FS により，REG-A=1 且つ REG-B=0 の場合，$f_{NTMW} \neq 0$ のとき f_{TW} を，それ以外の場合には 0 を出力し，REG-A 及び REG-B が上記以外の場合には f_{TW} を f'_{TW} として出力し，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．

上述した合成方式によれば，TW と NTMW それぞれに対して独立したメモリを設けることから，ウィンドウの移動，リサイズ，スクロール等のウィンドウ操作について，文献 [1] のマルチウィンドウシステムと同様の原理で高速に実行可能である．

3. ウィンドウ画像演算の実現法

3.1 透視化のためのハードウェア構造

図１の OPC-A 及び OPC-B におけるウィンドウ画像透視化機能を実現するためのハードウェア構造を図２に示す．COT は図１の COT に対応し，各ウィンドウ（背景ウィンドウも含む）と１対１に対応したエントリを持つメモリである．COT の各エントリには，透視化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．また，2'sCOMP は 2 の補数生成器，$MUX_1 \sim MUX_3$ はいずれも 2-1 マルチプレクサである．また，REG は図１の OPC-A においては REG-A，OPC-B においては REG-B に対応し，透視化を行う場合 0，行なわない場合 1 に設定されるレジスタである．さらに，MUL_1 と MUL_2 は乗算器であり，ADD は加算器である．マルチウィンドウ画像の透視化は，REG-A，REG-B で OPC 内部の乗算

表 1: マルチウィンドウ画像の透視化と重みの対象

透視化対象	REG(-A,-B)	透視化を施す OPC
非透視	(1,1)	none
NTMW&BG	(0,1)	OPC1
TW&NTMW	(1,0)	OPC2
ALL	(0,0)	OPC1&OPC2

器の稼動を制御することにより４通りに設定でき，ユーザの用途に応じた透視化画面を作成することができる．REG-A，REG-B の設定値と実現される透視化機能との関係を表１に示す．以下，ウィンドウ画像の透視化実現原理を述べる．

3.2 透視化を行わない場合

REG-A，REG-B 共に１に設定されている場合，透視化を行わずマルチウィンドウ画像の表示を行う．この時，画面の走査に同期して出力される各画素毎に，OPC-A においては，f_{NTMW} に基づいて COT から読み出された重みと i_{NTMW} とを乗じることにより，MUL_1 の出力として生成される．通常，透視化を行わないとき COT のエントリはすべて１に設定されるので，MUL_1 の出力は i_{NTMW} そのものとなるが，COT のエントリを１以下に設定すれば，対応したウィンドウの輝度値を低下することもできる．これと並行して，REG-A が１に設定されているので，MUX_1 からは COT の出力，すなわち，背景ウィンドウ画像がそのまま出力され，MUL_2 の出力には，背景画像が生成される．この際も同様に背景ウィンドウに対応したエントリを１以下に設定することにより，背景画像の輝度を低下した画像を出力することもできる．そして，MUX_2 により，MUL_1 と MUL_2 の出力が f_{NTMW} に基づいて適宜選択・出力され，(任意に輝度の低下が可能な) NTMW 画像が MUX_3 を介して i'_{NTMW} として出力される．OPC-B の構造は OPC-A と全く同様の構造であるが，MUL_1 の入力は i_{TW}，MUL_2 の入力は i'_{NTMW}，及び COT へ入力する領域情報は f'_{TW} となる．動作は OPC-A と同様であり，結局 i'_{NTMW} に i_{TM} を重ねたマルチウィンドウ画像を生成することになる．

3.3 透視化を行う場合

画像の透視化は，OPC-A においては i_{NTMW} と i_{BG} とを加重加算することにより実現する．i_{NTMW} に対する処理は，COT のエントリの内容が，透視化を行う際の重みに変わること以外，上述した輝度低下の場合と同様である．これに対して，i_{BG} に対する処理は，COT に格納される NTMW を構成する各ウィンドウの重みを用いて i_{BG} に対する重みを自動生成する点が異なる．ただし，COT の背景ウィンドウに対するエントリには０を格納する．また，OPC-B においては画像入力が i_{TW} と i'_{NTMW} に変わるのみで動作原理は同様であるのでここでは省略する．

OPC-A において透視化を行う場合，REG-A は０に設定されるので MUX_1 は”０”側の入力が選択・出力される．$f_{NTMW} \neq 0$ (i_{NTMW} として NTMW の画素がされている) の場合，COT から出力される重みの小数

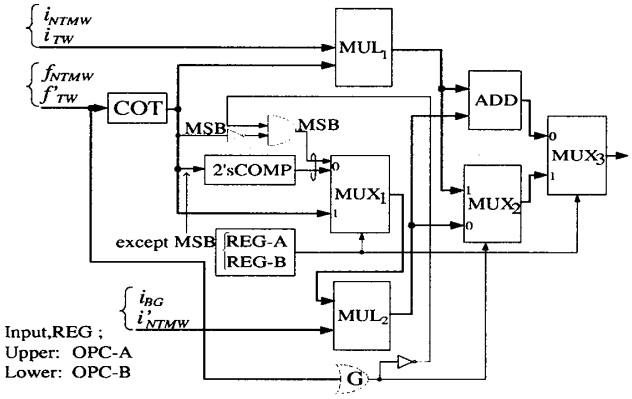

図 2: 輝度の低下と透視化の実現原理

部のみが $2'$sCOMP により２の補数表現に変換され，背景画像の重みの小数部となる．このとき，OR ゲート G (入力数は f_{NTMW}，f'_{TW} のビット数と同じ) の出力は１となるので，整数部 (MSB) は常に０となる．この結果，背景ウィンドウの画像の重みは，それと NTMW の重みとの和が常に１となるように決定されることになる．特に，COT の出力が１のとき，背景ウィンドウの画像の重みは，その MSB が０，小数部も $2'$sCOMP により０となり，結局０となる．$f_{NTMW} = 0$ の場合，COT からは０が出力される．このとき G_1 の出力は０となっているので，背景画像の重みの整数部が１となり，小数部は $2'$sCOMP により０となる．

以上のようにして生成した重みを用いて MUL_1 及び MUL_2 によりそれぞれ i_{NTMW} 及び i_{BG} に重み付けをし，ADD により加算することにより，透視化した画像を生成し，MUX_3 を介して i'_{NTMW} として透視化された NTMW 画像 i'_{NTMW} を出力する．OPC-B における REG-B が０に設定されている場合，上記 OPC-A と同様の手順により i_{TW} と i'_{NTMW} とを透視化したマルチウィンドウ画像を生成する．

上記のいずれの画像演算の場合においても，すべての画像メモリの内容自体は変化しない．したがって，透過化を解除する場合や，それらの度合いを変更する場合には，REG-A,REG-B 及び COT の内容を変更するだけで高速に実行できる．

4. あとがき

本稿では，高速にウィンドウを透視化できるマルチウィンドウシステムの実現法を与えた．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する予定である．

参考文献

[1] 佐藤洋一郎，横平徳美，籠谷裕人，岡本卓爾，茅野功 ：”描画時合成方式と表示時合成方式の併用によるスムーズ操作が可能なマルチウィンドウシステム”，信学論 (D-I)，vol.J86-D-I，no.9，pp.650-660，Sept. 2003.