K_067

透視化機能をもつマルチウィンドウシステムの高解像度化
High Resolution Multiwindow System with Ability of See-through of Windows.

幡谷 大介† 宮岡 有希† 宮崎 仁† 横川 智教† 茅野 功‡ 佐藤 洋一郎† 早瀬 道芳†
Daisuke Hataya Yuki Miyaoka Hisashi Miyazaki Tomonori Yokogawa Isao Kayano Youichiro Sato Michiyoshi Hayase

1. まえがき

近年，操作性，視覚効果を向上させるために，マルチウィンドウシステムにはウィンドウの透視化機能を標準で装備しつつある [1]．従来，透視化の機能はソフトウェア的に実現されており，処理性能の高いコンピュータを必要としていた．そこで，筆者らは，ウィンドウの透視化機能をハードウェア的に実現するマルチウィンドウ合成原理を提案し，それに基づくシステムの試作を行った [2][3]．他方，表示装置は高解像度化の一途をたどっており，これに伴ってマルチウィンドウシステムも高解像度への対応が求められる．しかし，本試作システムの解像度はVGA(横640 pixels，縦480 pixels)規格であり，高解像度化については何等検討していなかった．そこで本稿では，SXGA+(横1400 pixels，縦1050 pixels)までを対象として，上記システムの高解像度化について検討した結果を報告する．

2. マルチウィンドウ合成原理

ウィンドウの透視化機能をもつマルチウィンドウ合成原理を図1に示す [2]．W_1 が操作の対象ウィンドウ（TW）であり，W_2 と W_3 が操作の対象外ウィンドウ（NTMW）を構成する．構成要素の機能は以下の通りである．

- BM_{TW}：TW の完全な画像の格納用メモリ
- BM_{NTMW}：NTMW を合成した画像の格納用メモリ
- BM_{BG}：背景の完全画像の格納用メモリ
- FM_{TW}：TW の完全な領域（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）の格納用メモリ
- FM_{NTMW} ：NTMW を構成する各ウィンドウ（背景ウィンドウを含む）の領域の格納用メモリ（形状は FM_{TW} と同様）
- OPC-A,B ：ウィンドウの透視化処理のための演算回路
- FS ：マルチウィンドウ合成を行うための画像選択回路
- REG-A,B ：透視化のタイプを指定するレジスタであり，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．(通常表示の場合には 11 に設定する．)
- COT：透視化率 (0 より大きく 1 以下) を MSB が整数部，残りが小数部の形式で格納するためのメモリ (

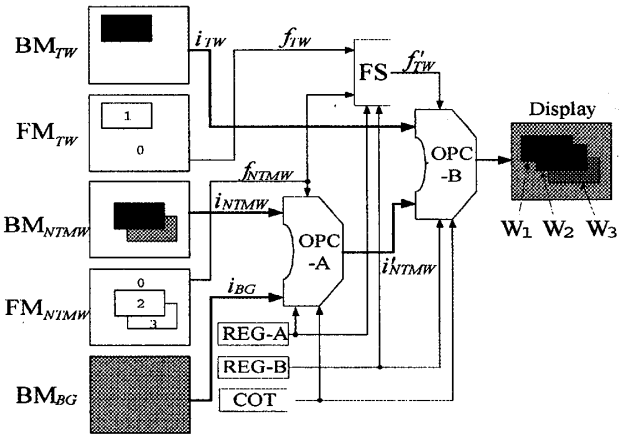

図 1: マルチウィンドウ合成原理

通常表示の場合全て 1)

表示画面上の k 行（垂直方向）l 列（水平方向）の画素の座標を (k, l) と記し，それに対応する BM_{TW} 及び FM_{TW} 上の情報を，それぞれ $I_{TW}^{(k,l)}$ 及び $F_{TW}^{(k,l)}$ と記す (その他のメモリ上の情報についても同様に記す)．また，ウィンドウ番号 n のウィンドウの透視化率（COT の内容）を $\alpha(n)$ と記す．

ここでは，紙面の都合上，TW，NTMW 双方を透視化する場合 (REG-A，B="00") についてのみ述べる．ドットクロックに同期して，BM_{TW}，FM_{TW}，BM_{NTMW}，FM_{NTMW} 及び BM_{BG} の情報は，それぞれ，シリアル信号 i_{TW}，f_{TW}，i_{NTMW}，f_{NTMW} 及び i_{BG} として読み出される．この際，(k, l) に対応するドットクロック周期に，$I_{TW}^{(k,l)}$，$F_{TW}^{(k,l)}$，$I_{NTMW}^{(k,l)}$，$F_{NTMW}^{(k,l)}$ 及び $I_{BG}^{(k,l)}$ が読み出される．OPC-A においては，

$$I_{NTMW}^{'(k,l)} = I_{NTMW}^{(k,l)}\alpha(F_{NTMW}^{(k,l)}) + I_{BG}^{(k,l)}(1-\alpha(F_{NTMW}^{(k,l)}))$$

にしたがって NTMW に対する透視化を施し，シリアル信号 i'_{NTMW} として出力する．さらに，OPC-B においては，

$$I_{TW}^{(k,l)} \times \alpha(F_{TW}^{(k,l)}) + I_{NTMW}^{'(k,l)}(1-\alpha(F_{TW}^{(k,l)}))$$

により，TW の透視化を施し，マルチウィンドウ画像を生成する．

†岡山県立大学 情報工学部
‡川崎医療短期大学 臨床工学科

3. 高解像度化のためのメモリ構成

3.1 メモリ構成

高解像度化に伴って，図1の各メモリに対しては，アクセスの高速化及び大容量化が求められることから，使用するメモリとして4 BANK構成のDDR SDRAMを想定する．ここでは，所要メモリ数の低減及び使用効率の向上を図るために，図1の各メモリを2つのBANKで実現し，2つのメモリ毎に1つのDDR SDRAMで実現する．ただし，M3のBANK CとDは未使用である．例えば，図1の各画像及び領域の情報を図2のように格納する．

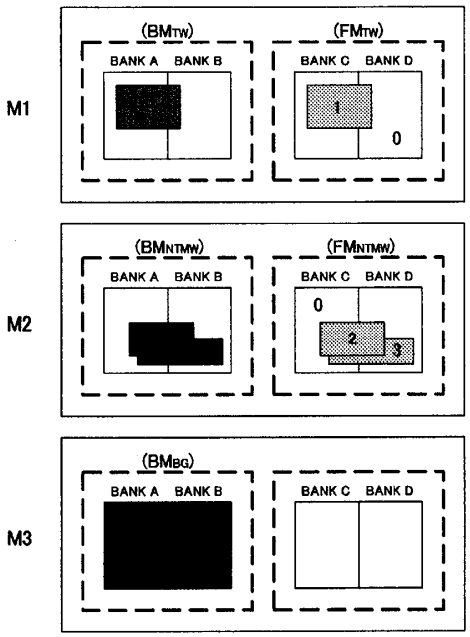

図 2: 画像情報と領域情報の格納状況

3.2 直列-並列変換処理

3.1のメモリ構成によれば，i_{TW} と f_{TW}（i_{NTMW} と f_{NTMW}）が直列に読み出されることになる．例えば，M1〜M3においてBANK A（B）とBANK C（D）から交互に，バースト長2で読み出しを行った場合，図3のようになる．最上部の信号はDDR SDRAMに印加されるクロックパルス（ドットクロックと同一周期）である．従って，図1の原理にしたがって透視化マルチウィンドウ画像を合成するためには，図4の（a）から（b）のような直列-並列変換処理が必要となる．

直列-並列変換処理の原理を図5に示す．M1から出力される直列信号 $I_{TW}^{(k,l-1)}$，$I_{TW}^{(k,l)}$，$F_{TW}^{(k,l-1)}$，$F_{TW}^{(k,l)}$ を順次シフトレジスタSR1に格納する．このとき，$S3 = I_{TW}^{(k,l-1)}$，$S1 = F_{TW}^{(k,l-1)}$ となっており，マルチプレクサMUXを介して並列信号 i_{TW} 及び f_{TW} として出力する．そして，ドットクロックに同期してシフトすることにより，$S3 = I_{TW}^{(k,l)}$，$S1 = F_{TW}^{(k,l)}$ となり，並列信号に変換できる．この処理と並行して出力される $I_{TW}^{(k,l+1)}$〜$F_{TW}^{(k,l+2)}$ は下段のSR2に格納し，MUXを切り替えてSR1と同様に出力すれば，図4（b）のような並列信号が生成できることになる．

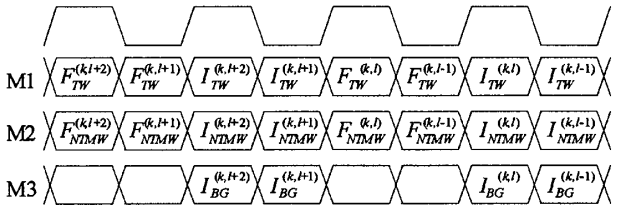

図 3: DDR SDRAMからの読み出し動作

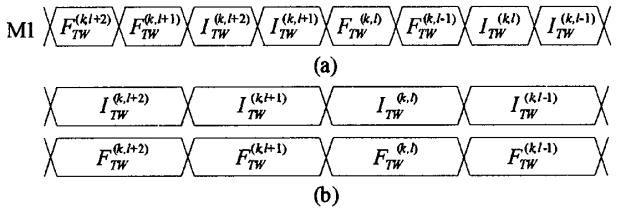

図 4: 直列-並列変換

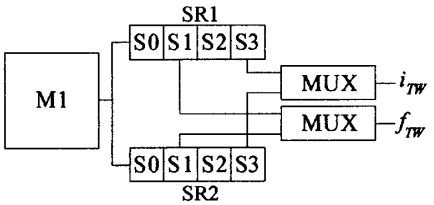

図 5: 直列-並列変換の原理

4. あとがき

本稿では，ウィンドウの透視化機能をもつマルチウィンドウシステムの高解像度化のためのメモリ構成について述べた．現在，透視化処理部（OPC-A，B）の高速化について検討している．

参考文献

[1] http://www.microsoft.com/japan/windowsvista/features/default.mspx

[2] 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，C-019, pp.265-266(2004-09)

[3] 宮崎仁，茅野功，佐藤洋一郎，横川智教，早瀬道芳:"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"，FIT2005論文集，C-024, pp.237-238(2005-09)