

シングルフレームバッファに適したビデオ処理方式

1 N-5

滝澤 哲郎 前野 和俊 福田 公彦
 NEC C&C 研究所 NEC

1 はじめに

近年、マルチメディアと称してビデオをPC/WS上で再生する应用が増え、その再生品質を向上させるためにビデオ処理専用ハードウェアを用いたり、グラフィックスアクセラレータにビデオ描画機能を付加する解が取られるようになってきている。これらのシステムでは、低コストが要求されることから、シングルフレームバッファでグラフィックスもビデオも表示できることが必須となっている。

シングルフレームバッファでビデオ表示を行う場合、いくつかの問題点があるが、本稿では2点の問題を取り上げる。一つは、複数のビデオを同時に表示する際にビデオ同士あるいはビデオとグラフィックスの領域の重なりが発生するが、これを処理する問題。もう一つは、ビデオの内容の書き換えとディスプレイへの表示の周期が異なることから、ビデオを書き換える過程が表示されてしまうという問題である。

筆者らは、ビデオ処理機能を備えたグラフィックスアクセラレータを開発するにあたり、上記の2問題を解決する手段を開発、試作を行なった。本稿では、その解法の詳細について述べる。

2 LSIの概要

筆者らが試作したLSI、MEGA (Multimedia Engine with Graphics Accelerator)はその名が示す通り、グラフィックスアクセラレータとマルチメディアエンジン(ビデオ処理エンジン)を一体化したLSIで以下の機能を備える。

- 2D (GUI) アクセラレータ
- 3D アクセラレータ
- ビデオデコードアクセラレータ

ビデオデコード処理の後処理にあたるフォーマット変換、色空間変換、拡大/縮小、描画の各処理をハードウェア化し、CPU単独で全処理を行なった場合と比べ、性能を飛躍的に向上させる。上記の後処理は演算量と比較してメモリアクセスの占める割合が非常に高く、CPU向きの処理とは言えない上、CPU単独では実行時間比で全処理の約半分を占め、CPUに大きな負担をかけて

いた。ハードウェア量に対する性能向上の効率が非常に高いため、この機能を搭載するグラフィックスアクセラレータが増えてきている。

• ビデオキャプチャ機能

アナログビデオ信号をデジタル化し、フレームバッファに描画するとともに、静止画または動画データとしてメモリに取り込む機能である。

• ビデオ表示制御機能

本稿で取り上げるシングルフレームバッファにおける課題に対応した機能を備える。一つは、複数の独立したビデオを同時表示するマルチビデオ表示機能。もう一つは、ビデオ表示における胴体切れを防止する描画/表示同期制御機能である。

3 マルチビデオ表示方式

マルチビデオ表示では、複数のビデオを任意の大きさ、位置に任意の形状で同時に表示することが要求される。

ビデオを一つだけ表示する場合には、表示するビデオの大きさ、あるいは表示画面全体の大きさを持つビットマップ領域(マスクプレーン)を用意し、その各点にビデオの表示/非表示の意味を持たせ、ビデオをフレームバッファに描画する際に参照するという方法が取られる。

これを複数のビデオ表示に対応させるために筆者らが取った方法は、このビットマップによるマスクプレーンを多ビットに拡張して、表示画面の各点に対して他ビットの情報を持たせたIDプレーンを用意するという方式である(拡張IDプレーン方式)。

これに対して、ビデオ毎に固有のマスクプレーンを持ち、個別に管理する方法も考えられるが、拡張IDプレーン方式の方が、以下の点で有利である。

- ビデオの大きさに係らず使用メモリ量が常に一定であり、IDプレーンのアドレス計算が簡単
- ビデオの枚数が多い場合にはメモリ量も少ない

拡張IDプレーン方式では、あらかじめ表示画面と同面積のIDプレーンを用意し、IDプレーンのビデオが表示される領域にはビデオ固有のID値を格納しておく(図1の1および2)。ビデオをFRBに描画する際に、IDプレーンに格納されている値と描画するビデオのIDとを比較し、一致したピクセルにだけ描画を行なう。これにより、ウィンドウはもちろん任意の形状の複数のビデオを同時に管理することができる。

MEGA では、2ビットのIDプレーンをフレームバッファの非表示領域に確保し、3枚までのビデオが同時表示可能である (ID=0は、グラフィックス用の領域を示し、すべてのビデオの描画を禁止する)。

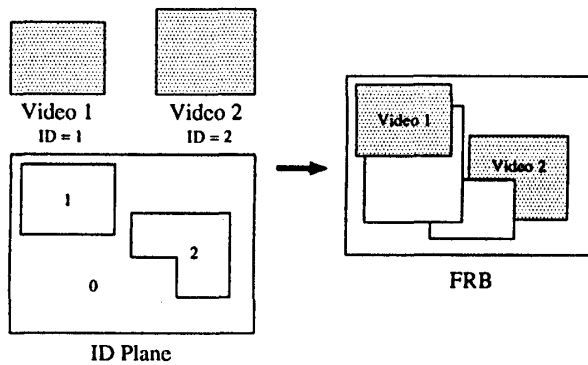


図 1: 拡張 ID プレーン方式

4 描画 / 表示同期制御方式

シングルフレームバッファでは、ビデオデータの描画速度と表示走査速度に差がある場合、ある瞬間にはビデオ映像の上部と下部とで異なるフレームが表示されていることがある。動きの激しい映像の場合はその内容の差が大きいため、映像が途中で横に切られたように見える (胴体切れ)。

筆者らはこの問題を解決するために、フレームバッファへのビデオデータ描画とディスプレイ表示のためのフレームバッファからのデータ読み出しのタイミングを同期させ、胴体切れを発生しにくくする方式を開発した (描画 / 表示同期制御方式)。

描画 / 表示同期制御方式では描画開始禁止範囲を設定し、表示走査線がその範囲にあるときはビデオ描画の開始を遅らせる制御を行なう。描画開始禁止範囲は、表示走査線がその範囲にあるときにビデオの描画を開始すると、ビデオ表示領域内で走査線と描画位置の追い越しが起きるといふもので、ビデオの表示位置、大きさ (ビデオデータの描画速度に相当する) から計算で求まる。

具体的には、次のようになる。まず、走査線位置の禁止範囲を表す 2 点の情報禁止開始位置 (Start) と終了位置 (End) を以下の方法により算出する (図 2)。

$$\frac{V}{width} < S \text{ のとき}$$

$$Start = y - \left(\frac{S \times height \times width}{V} - height \right)$$

$$End = y$$

$$\frac{V}{width} > S \text{ のとき}$$

$$Start = y$$

$$End = y + \left(height - \frac{S \times height \times width}{V} \right)$$

V: ビデオ描画速度 (pixel/s)
 S: 表示走査線速度 (line/s)
 y: ビデオ表示開始垂直位置
 width: ビデオ表示幅
 height: ビデオ表示高

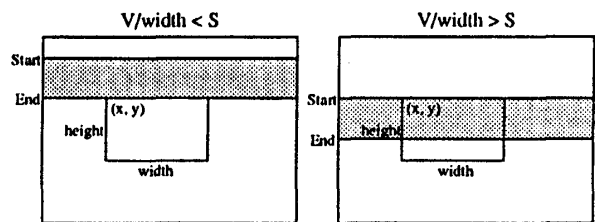


図 2: 描画 / 表示同期制御の禁止範囲

そして、走査線位置 (Scan) と禁止範囲との関係が、

Start < End のときは

$$Start < Scan < End$$

Start > End のときは

$$Start < Scan \text{ OR } End > Scan$$

を満たしているかどうか判定し、前記の条件が成立しなくなるまで描画開始を遅らせることで、ビデオの描画領域内では表示走査線位置と描画位置の追い越しが起きないようにすることができる。

ビデオの描画速度が一定しない場合には必ずしも胴体切れを防止できるとは限らないが、描画速度の上 / 下限がわかっているならば禁止範囲をそれぞれについて求め、その論理和を取ることで胴体切れの防止を保証することができる。

ただし、描画速度が表示走査速度に比べて極端に遅い場合 (ビデオの大きさが大きい場合) は禁止領域が画面全体を超えるため、胴体切れを防止することはできない。

5 おわりに

本稿では、シングルフレームバッファでグラフィックスとビデオを表示するシステムにおけるビデオ表示処理の課題を 2 点取り上げ、その解法について述べた。

この他にも、シングルフレームバッファに起因することではないが、PC/WS におけるビデオ表示の課題として、拡大 / 縮小処理におけるフィルタの高画質化やディザに代表される減色処理などがあげられる。特に高画質の減色処理の実現は、フレームバッファのメモリ量を減らし、システムの低コスト化に果たす役割が大きい。