

分散型グラフィックスシステムの構成に関する検討

6N-5

大金 顕二 田中 英彦 元岡 達

( 東京大学 工学部 )

1. はじめに

高性能のグラフィックスデバイスの出現や、分散処理技術の発展などにより、コンピュータグラフィックスの分野に於いても、今後、分散型のシステムの利用が多くなると考えられる。本報告では、分散型グラフィックスシステムの構成について、グラフィックスパッケージのスタンダードとの関連を中心に述べる。

2. 分散型グラフィックスシステム

2-1. 分散型システムの構成

分散型グラフィックスシステムのモデルとして、図1のような、グラフィックスデバイスとミニコン又はマイコン程度のローカルプロセッサが接続され、さらにそれが通信回線を介してホストの大型コンピュータに接続されている構成を考える。

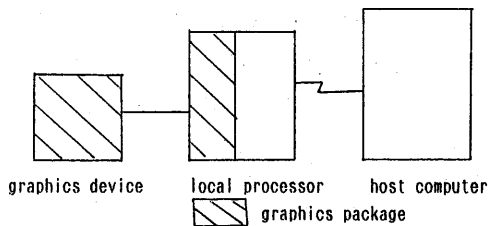


図1 分散型システムのモデル

このような分散型システムの利点として、単純な応答のローカル処理によるターンアラウンド時間の短縮、大規模計算のホストによる高速計算、ホストのデータベースや各種ライブラリの使用、特殊リソースの共用による効率化、機能のハードウェア化による高速処理などが挙げられる。

グラフィックスのソフトウェアパッケージとして、最近、コアシステムやGKS等のグラフィックスパッケージのスタンダード案が目立っているが、ここではコアシステムを考えることにする。分散システムに於けるコアシステムの配置は図1のように、デバイスとローカルプロセッサに分散させることが考えられる。この場合、デバイス側に分散される機能は主に、ハードウェア化、ファームウェア化が効率的に行なわれる機能が中心となる。

2-2. アプリケーションプログラムの構成

アプリケーションプログラム (AP) の一般的な構成は、図2のようになっている。ローカルプロセッサにはAPのメインルーチンがあり、簡単なインタラクティブ処理や図形の変更はローカルで行なう。モデリングシステムやDBMS等の大きく複雑な処理は、ホスト側にライブラリとしてサポートされており、ローカルのAPからそのライブラリを呼び出す形でアクセスされる。また、ユーザが記述した大きな処理を行なうAPも、一まとまりのパッケージの形でホスト側におかれ、他のライブラリと同様の呼び出し方で使用される。

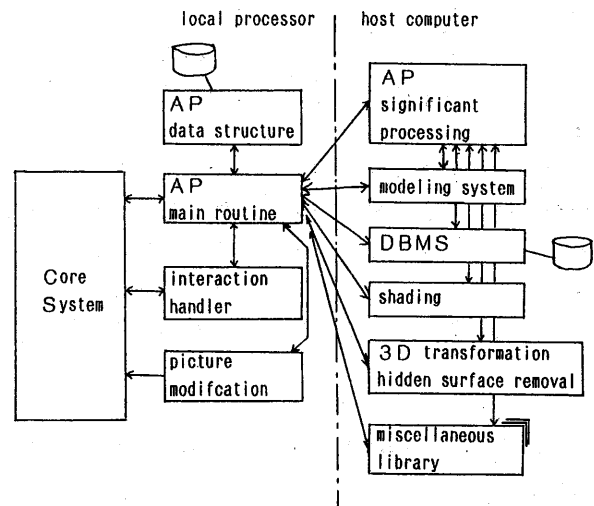


図2 アプリケーションプログラムの構成

3. 実験システム

当研究室では、ミニコンMS50にカラーイメージディスプレイTGD612が接続されたシステムがあり、これにコアシステムが実装されている。このコアシステムは表1のようなレベルをもつもので、ラスタエクステンションも実現されている。

	level
output	bufferd
input	synchronous
dimension	2D
hidden-surface	explicit

表1 コアシステムのレベル

分散システムの実験システムとして、当システム

を通信回線を通じて、東大大型計算機センターのM280Hと接続し、M280H側にライブラリとして3次元変換、及び、隠面消去を行なうパッケージを実装する。以下、このパッケージについて説明する。

#### 4. 3次元パッケージ

##### 4-1. コアシステムと隠面消去

コアシステムには、隠面消去をサポートするレベルがあるが、隠面消去は、ハードウェアのサポートがない場合、小さなローカルプロセッサでは処理負荷が過大であると考えられる。したがって、必要に応じてホスト側のライブラリとして隠面消去プログラムが使えることが望ましい。この場合、隠面消去は3次元変換後に行なわれるため、ホスト側では3次元変換も含めたライブラリとなる。以上のようなライブラリを3次元パッケージと呼ぶことにする。

##### 4-2. アクセス方法

3次元パッケージへのアクセスの方法としては、できるだけ通常のコアシステムの関数の指定方法に準拠したものが望ましい。つまり、アプリケーションプログラムの記述は通常3次元コアシステムの場合とほとんど同様の記述を可能にし、プログラムポータビリティを保存する。3次元パッケージへのアクセス方法として次の方法が考えられる。

- ①ローカル処理とホスト処理の関数の名称を区別する。
- ②ホストへのアクセスはエスケープ機能を使う。
- ③コアシステムの状態によってシステムが振り分ける。

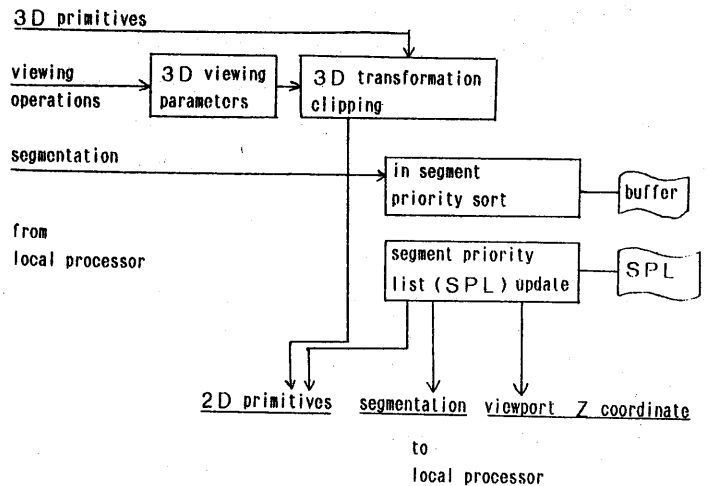
このうち③が理想的であるが、サブルーチンのコール機構はOSに依存する場合が多く、実現が困難なことがある。コアシシステムでは3次元の出力要素と2次元の出力要素の関数名は区別されている。したがって、ここでは、出力要素は、3次元の出力要素をホストに転送し、他の関数（create\_segment等）でホストに転送する場合は①の方法を用いることにする。ただし、この方法では、2次元と3次元の出力は完全に分離される。

##### 4-3. 隠面消去アルゴリズム

隠面消去はNDCスペースで行なわれるためオブジェクトスペースのアルゴリズムで代表的なプライオリティアルゴリズムを採用する。ここで、ローカルでのコアシシステムは、隠面消去レベルがexplicitであり、面の優先順位はセグメントの優先順位、つ

まり、viewportのZ座標としてコアシシステムに通知される。このとき問題となるのが、セグメント間で優先順位のオーバーラップがある場合であるが、ここでは優先順位のオーバーラップはないものとする。セグメント間の優先順位のオーバーラップがある場合も、セグメントの分割により解決できるが、その場合セグメントの管理がホスト側に移行し、ホスト、ローカル間の通信量が増大する問題点がある。

図3 3次元パッケージの構成



##### 4-4. 3次元パッケージの構成

3次元パッケージの構成は図3のようになる。3次元変換された出力要素は、隠面消去が行なわれない場合はそのままローカルに出力され、隠面消去される場合は、バッファに蓄えられ、セグメントがクローズされると、セグメント内でプライオリティソートが行なわれる。3次元パッケージでは、セグメントの優先順位とセグメント内で最も優先順位の高い面の情報を保存したリスト（SPL）を保持し、新しいセグメントは、SPLの適当な場所に挿入され、プライオリティソートされた出力要素は適当なviewportのZ座標と共にローカルに出力される。

#### 5. おわりに

本報告では、分散型グラフィックスシステムの、ローカルプロセッサとホスト間の分散処理について、特に3次元パッケージに関して、コアシシステムとの関連を中心に述べた。現在、実験システムで実装中であるが、今後は、このシステムの評価、機能拡張、他のライブラリとの関連などが検討課題である。

##### 〈参考文献〉

“Status Report of the GSPC“,  
Computer Graphics, vol. 13, no. 3, Aug.  
1979.