

# 分散型グラフィックスシステム

大金 顕二, 田中 英彦, 元岡 達  
(東京大学 工学部)

## 1. はじめに

最近のハードウェアコストの低減や、分散処理技術の発展などにより、今後コンピュータグラフィックスの分野に於いても分散処理が様々な形で導入されると考えられる。グラフィックスシステムに於ける分散処理には大きく分けて2種類のものがある。1つはグラフィックス装置に固定的に、比較的高度なグラフィックス機能を付加するものである。他の1つはグラフィックス装置が接続されたローカルコンピュータと通信回線やLAN経由で結合されたホストコンピュータによる分散処理で、アプリケーションプログラマが機能の分散をある程度制御できるものである。前者の形はインテリジェントグラフィックスターミナルとして最近よくみられるが、今後は後者の形も多く利用されると考えられる。

本研究では、この後者のタイプを中心に、分散型グラフィックスシステム(分散型図形表示システム)について、実際に実験システムにおいて実装することにより、そのグラフィックス機能の分散やシステムの構成法、及びコアシステムとの関係を検討する。

## 2. 分散型図形表示システムの一般論

### 2-1 分散型図形表示システムの特徴

分散型図形表示システムは、複数の計算機やグラフィックス装置にグラフィックス機能を分散させ、高性能で効率のよい図形表示を行なうシステムである。ハードウェア構成は図1(a),(b)のようなものである。このようなシステムの利点として次のようなものが挙げられる。

① ローカル側の計算機や装置はユーザ

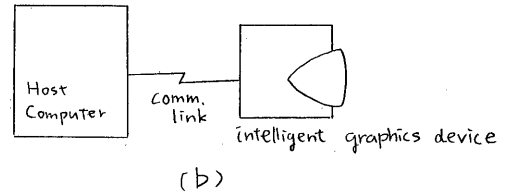
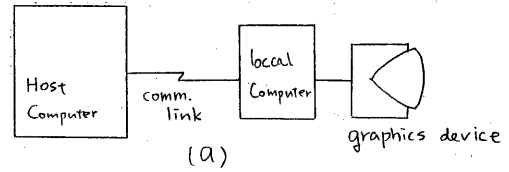


図1 分散型図形表示システムの構成

- の身近にあり、使いやすい。
- ② 簡単な処理はローカルで行ないホストの負担が軽減できるとともに通信量を減少させることができる。
- ③ 特殊なリソースはホスト側で共用することができる。
- ④ 簡単な応答処理をローカルで吸収することによって速いレスポンスが得られる。
- ⑤ 経済性が向上する。

また、このようなシステムの構築にあたっては、(i)ハードウェア、ソフトウェアそれぞれの機能の分割法、(ii)プロセス間通信機能の実現方法、(iii)ユーザへの使いやすいプログラミング環境の提供、を検討する必要がある。

### 2-2 分散型図形表示システムの種類

分散型図形表示システムには次の2種類がある。

#### ① 機能固定型システム

これは主に図1(b)の形をとり、グラフィックス装置に固定的に機能を分散させたインテリジェントグラフィックス装置を用いたシステムである。装置に分散させる機能としては、入力に対

する簡単な応答，入力と密接な関係がある処理，ディスプレイファイルの保持，図形変換，クリッピング等がある。

### ② プログラマブル型システム

これは主に図1(a)の形をとり、ローカル側にグラフィックスパッケージの機能があり、応用プログラムの分散により機能の分散を行なう。応用プログラムの主体はローカル側にあり、ホスト側は大規模な計算を高速処理する場合、特殊リソースの利用，ライブラリの利用のときに使われる。プログラマブル型システムに於いては応用プログラムのプログラミング環境が重要で、応用プログラムをホストとローカル間の通信の詳細を意識することなく記述でき、プログラムポータビリティを保つ必要がある。

従来は①の形の研究が多く、②のシステムの研究も一般的に構成法についての研究が多かった。(4) 本研究では②のシステムについて、具体的な応用を通じて、機能の分割法，コアシステムとの関係を中心に検討する。

## 3. MT-COREの構成

### 3-1 実験システムの構成

実験システムのハードウェア構成は図2のように、ミニコンMS50とカラーイメージディスプレイTGD612、及び東大大型計算機センターから成る。TGD612は解像度が512×512ドット、

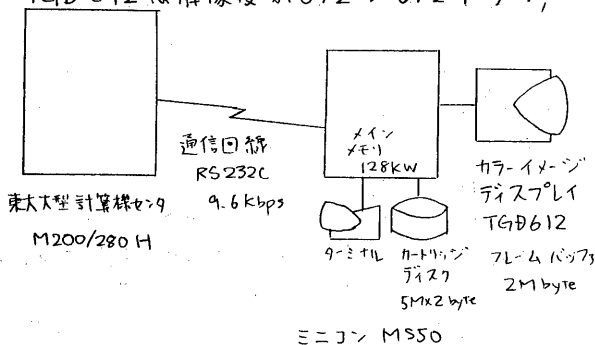
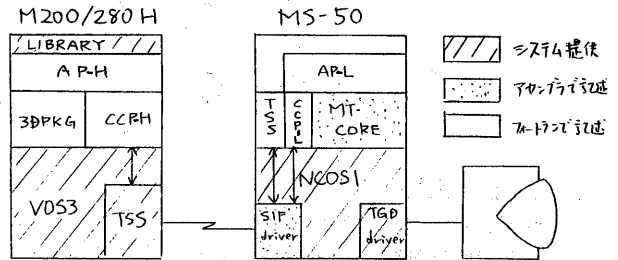


図2 実験システムのハードウェア構成



- 3DPKG : 3-DIMENSIONAL PACKAGE
- CCRH : COMMUNICATION CONTROL PROGRAM AT HOST
- CCPL : COMMUNICATION CONTROL PROGRAM AT LOCAL
- APH : APPLICATION PROGRAM AT HOST
- APL : APPLICATION PROGRAM AT LOCAL

図3 実験システムのソフトウェア構成

同時表示カラー数2<sup>24</sup>色のラスタスキャン型のディスプレイである。実験システムのソフトウェアは図3のような構成で、MS50上のグラフィックスパッケージとしてコアシステム準拠のMT-COREをインストールした。

### 3-2 コアシステム

グラフィックスソフトウェアの装置依存性を解消するために、装置依存部はグラフィックスパッケージで吸収し、応用プログラムは装置独立なものにすることが考えられた。このようなグラフィックスパッケージを装置独立なグラフィックスパッケージと呼び、コアシステムはこの標準化案の一つである。(5) これにより応用プログラムのポータビリティが実現できる。コアシステムの設計思想は synthetic camera analogy と呼ばれ、図形の表示を、物体の写真を撮ることに対応づけている。これによりコアシステムではモデリング変換とビューイング変換の分離を明確に規定している。

### 3-3 MT-COREの特徴

本研究でインストールしたMT-COREはコアシステム準拠のグラフィックスパッケージ

ージで、アセンブラで約8500行、コードにして約25Kワードであり、現在フォートランとLispから利用できる。MT-COREの実装レベルは、保持型セグメント、同期入力、2次元図形、セグメントの優先度指定をサポートするレベルである。MT-COREの機能には、出力要素(直線、多角形、文字)の定義、セグメントの処理、属性の操作、ビューイング変換、入力機能、制御機能がある。セグメントは出力要素の集合でデータ取扱いの単位であり、属性は出力要素の色や形状等の性質を規定するものである。ビューイング変換はユーザ座標系から装置座標系への変換である。入力機能としてはジョイスティックによる入力をサポートしている。

MT-COREの内部構成は図4のようになっている。内部はディスプレイファイルまでの装置独立部と、それ以降の装置依存部(デバイスドライバ)に

MT-CORE

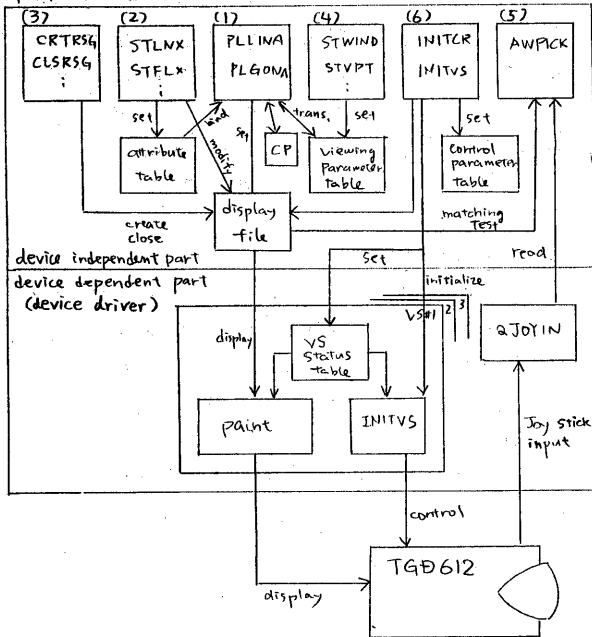


図4 MT-COREの内部構成

分けられる。多角形の表示はソフトウェアでサポートしており、基本的にはY-Xアルゴリズムによりトスキャンライン塗り潰している。

#### 4. 分散型図形表示システムの実装

##### 4.1 分散型図形表示システムのモデル

本研究で考えているモデルはプログラマブル型の分散型図形表示システムであり、ハードウェア構成は図1(a)のようなものである。ローカル側のコンピュータはミニコンまたはマイコン程度の使いやすいユーザ専用計算機で、これにグラフィックス装置が接続されている。グラフィックス装置はラスタスキャン型を主に考え、ローカルコンピュータと一体化したものも考えられる。ホスト側は大型計算機、あるいは計算機センタである。ローカルとホスト間の通信回線は付加ハードウェアをあまり必要としない一般的なものが望ましい。

ソフトウェア構成は図5のようになっている。グラフィックスパッケージは標準化されたものがローカルにあり、ホスト側にはシステムが提供する豊富なライブラリが備わっている。ユーザ

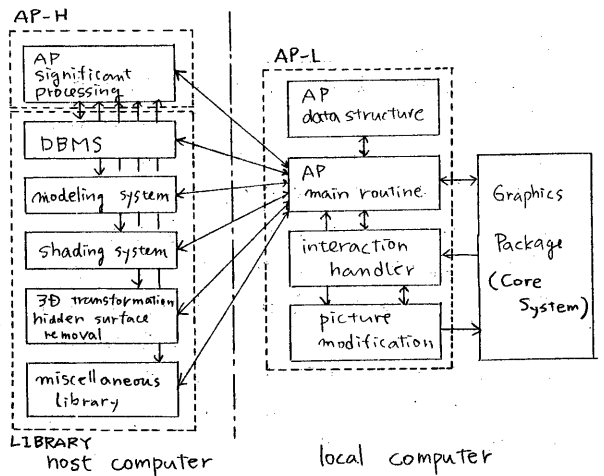


図5 分散型図形表示システムのソフトウェア構成

が記述したアプリケーションプログラムはローカルとホストに分散させる。その主体はローカルにあり、ホスト側のアプリケーションプログラムは大規模な計算やライブラリの利用、特殊リソースの利用のときに使われる。アプリケーションプログラムからはサブルーチンコールによってグラフィックスパッケージやホスト側のライブラリ、アプリケーションプログラム等を利用する。想定される各プログラム間のサブルーチンコールには図6のような種類が考えられるが、このうち実験システムで許しているのは①②③④⑦である。⑦を許すことにより、後述するようにアプリケーションプログラムのポータビリティを保ち、分散のための制御をアプリケーション内に記述する必要がなくなる場合がある。

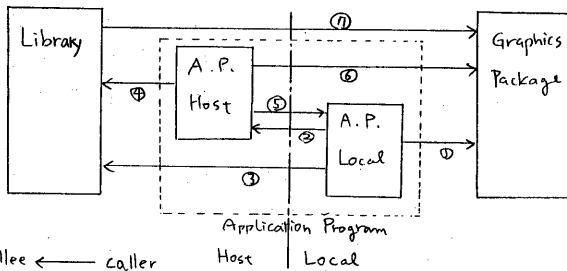


図6 アプリケーションからのライブラリ、グラフィックスパッケージの利用

#### 4-2. 3次元パッケージ

3次元図形処理における隠面消去処理はプライオリティアルゴリズムのような方式をとれば、グラフィックスパッケージの外で実現可能であり、ホスト側のライブラリとして配置することができる。そこで実験システムでは、ホスト側に3次元変換と隠面消去を行なうライブラリを作成し、分散型図形表示システムの実験を行なった。以後このライブラリを3次元パッケージ(3DPKG)と呼ぶ。

3DPKGはフォートランから呼べるサブルーチンパッケージで、フォートランで約2900行、53個のサブルーチン

ンから成る。3DPKGとアプリケーションプログラムのインタフェースは3次元のコマンドシステムに準拠している。これは図7に示すように図6の⑦のサブルーチンコールを許したためであり、これによりアプリケーションのポータビリティが保たれる。

⑦ 禁止

Call polygon\_3(X,Y,Z,N,XR,YR) ← 3DPKG call  
Call polygon\_2(XR,YR,N) ← MT-CORE call

⑦ 可能

Call polygon\_3(X,Y,Z,N) ← 3DPKG call  
内部で MT-CORE call

図7 3DPKG→MT-COREのサブルーチンコールを許したときのアプリケーションの相関

3DPKGの機能には多角形や直線等の出力要素定義機能、ビューイング変換として3次元の透視、平行投影、ウィンドウと深さによるクリッピング機能および隠面消去の機能がある。3DPKGの内部処理としては、ビューイング変換と隠面消去が中心となる。隠面消去はプライオリティアルゴリズムを利用した。これは面(多角形)に、ある基準で優先順位をつけ、これが低いものから順に描いていくものである。

#### 4-3. 通信制御

通信の制御のためにマシン間のサブルーチンコールを実現するプログラムCCP-HとCCP-Lを作成した。これは図

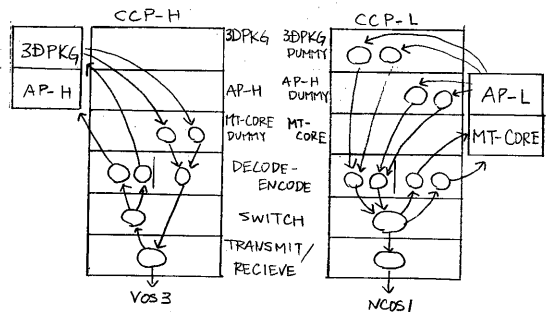


図8 CCP-H, CCP-Lの構成

8のような構成になっている。相手側計算機のサブルーチンを呼び出したときは、そのダミーサブルーチンへ制御が移り、次に引数を転送用に変換するルーチン、実際に転送するルーチンへと進む。受信側は転送データの先頭のコードを調べ、サブルーチンを決定しデータを内部表現に変換してサブルーチンコールをする。

## 5. 分散型図形表示システムの評価

### 5-1. プログラムポータビリティ

分散型図形表示システムの実験システムでは応用プログラムからサブルーチンコールの形でホスト側の3DPKGが利用でき、通信の詳細を意識する必要がない。また、3DPKGと応用プログラムのインタフェースを3次元コアシステムと同様にしたため、プログラムのポータビリティが保たれる。しかし、MT-COREと3DPKGのサブルーチンを明確に区別したことにより、3次元と2次元の出力要素や、セグメント、属性操作機能が分離されてしまう。逆に、MT-COREと3DPKGを1つの3次元コアシステムとした場合、分散を意識する必要はないが、制御が複雑になり実行時のオーバヘッドが増加する。また、システムの状態テーブルの管理等のため通信量が多くなってしまった。したがってホスト側の3次元処理ルーチンは、ここで作成したようにライブラリとして扱う方が効率がよいと考えられる。

### 5-2. 処理時間の評価

分散型システムとスタンドアロン型システムとの処理時間を、例を用いて比較した。例は複数の四角錐を描くプログラムである。ここでは図形の複雑度を面(多角形)の数で代表させ、面の数と処理時間の関係、を求めた。分散型システムの処理時間( $t_d$ )とスタン

ドアロン型システムの処理時間( $t_e$ )を図10に示す。また、 $t_d/t_e$ を図11に示す。 $t_d < t_e$ となるのは次のとおりである。

回線の伝送速度	$t_d < t_e$ となる $n, t_e$
9.6 Kbps	$n > 2320$ 個, $t_e > 1700$ sec
48 Kbps	$n > 360$ 個, $t_e > 230$ sec
1 Mbps	$n > 0$ 個, $t_e > 0$ sec

このとき処理時間のうち支配的であるのはホストからローカルへの多角形情報の転送と、それに続くローカルでの多角形表示時間である。したがって図9のようにデータの転送と表示を並

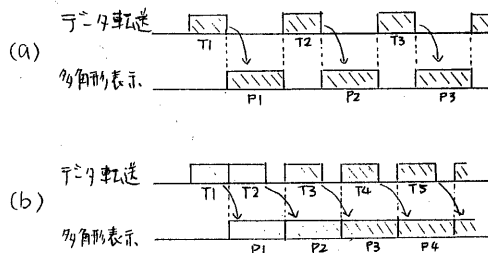


図9 表示とデータ転送の並列化

列化した場合、処理時間の短縮が期待できる。このとき $t_d < t_e$ ,  $t_d/t_e$ は図12, 図13のようになり、 $t_d < t_e$ となるのは次のとおりである。

回線の伝送速度	$t_d < t_e$ となる $n, t_e$
9.6 Kbps	$n > 28$ 個, $t_e > 20$ sec
48 Kbps	$n > 7$ 個, $t_e > 5.6$ sec
1 Mbps	$n > 0$ 個, $t_e > 0$ sec

以上のように、比較的低い伝送速度でも分散型システムの方がスタンドアロン型システムよりも処理時間が短くなる場合が多くなる。現在、実験システムはこの方式をとっている。

### 5-3. 機能の分散

実験システムでは3次元図形処理は

ホスト側で行ない、2次元図形処理はローカルで行なうように機能を分散している。処理時間に関しては前節で述べたように分散型システムの方がスタンドアロン型システムよりも有利になっている。また、メモリ容量を考慮

と、3次元処理をローカルに配置した場合にはメモリ容量の制限から図形の複雑度に制限を加える必要も出てくる。3次元図形処理がホスト側にある場合は、ホスト側でモデリングシステム等の上位ルーチンで統一的に処理を行な

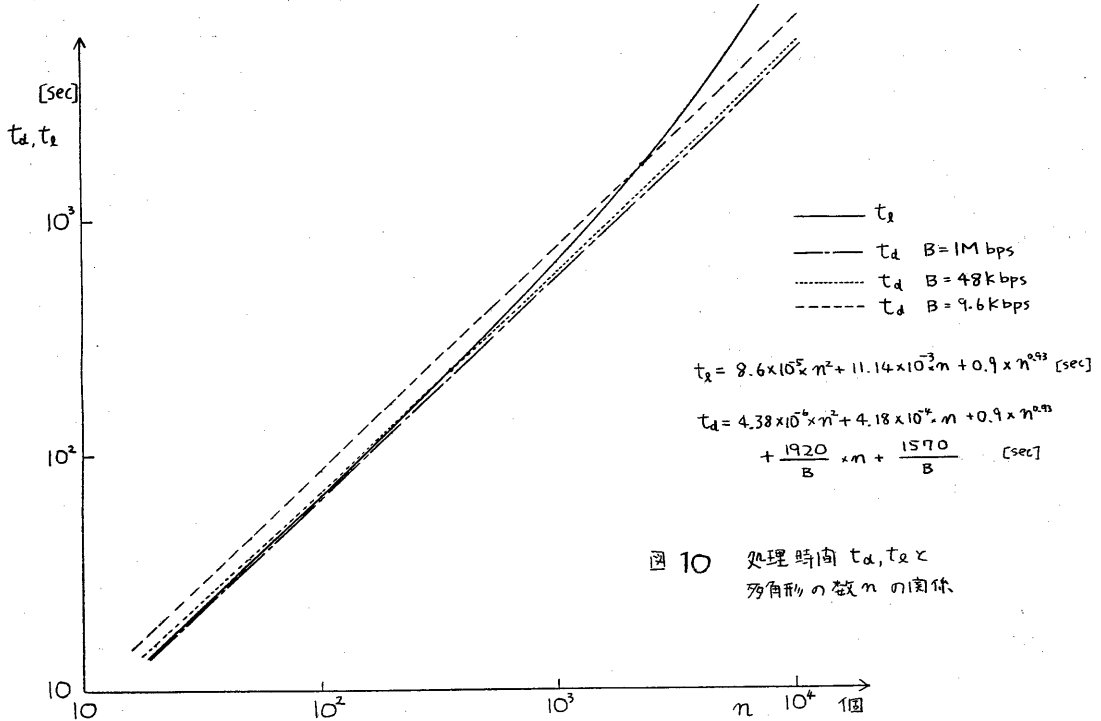


図 10 処理時間  $t_d, t_e$  と多角形の数  $n$  の関係

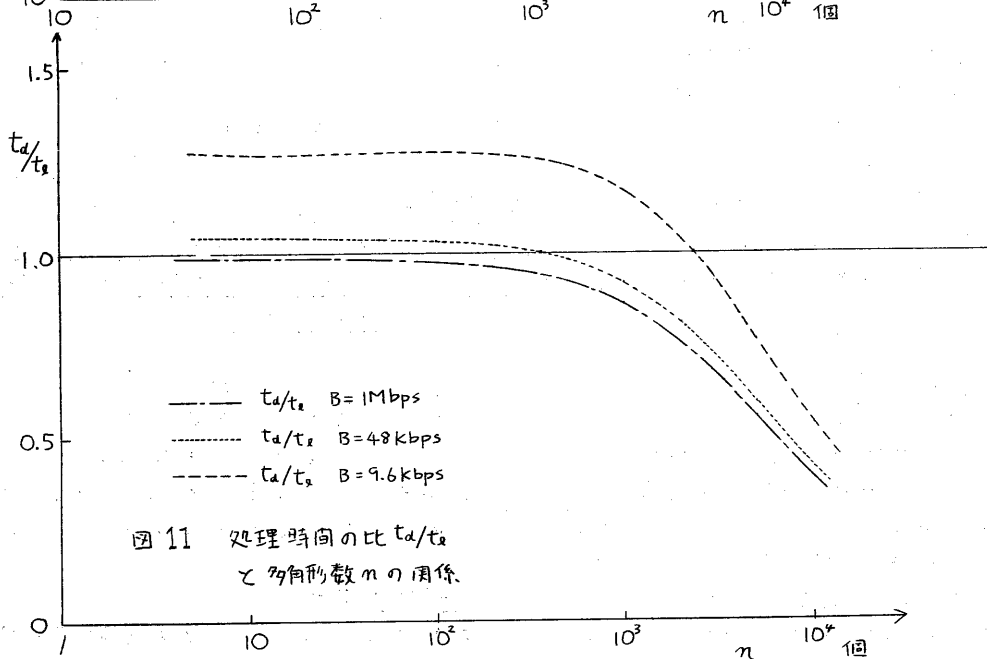


図 11 処理時間の比  $t_d/t_e$  と多角形数  $n$  の関係

うことも可能である。以上の理由から3次元図形処理をホスト側に配置することは妥当であるといえる。

5-4. 実験システムの利用環境

現在、実験システムではセンタ側の

3DPKGを利用する限りはアプリケーションプログラムは通信の詳細を記述する必要がない。しかしユーザ記述のアプリケーションプログラムをホスト側に配置する場合は、図10で述べたような通信制御部をユーザが記述する必要がある。そ

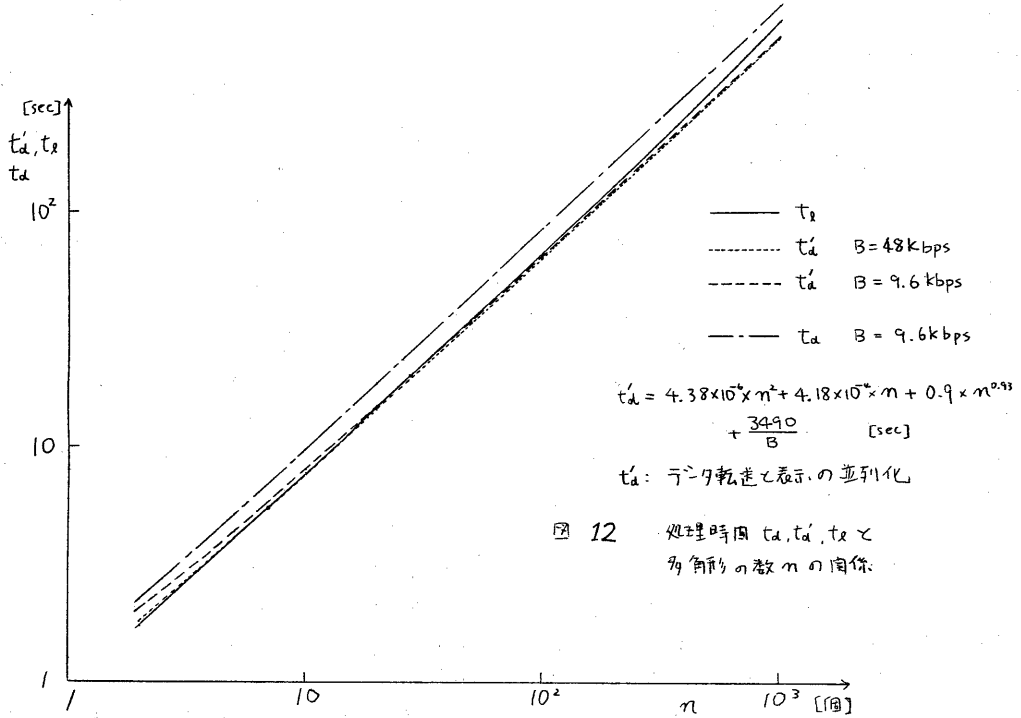


図 12 処理時間  $t_d, t_d', t_e$  と多角形の数  $n$  の関係

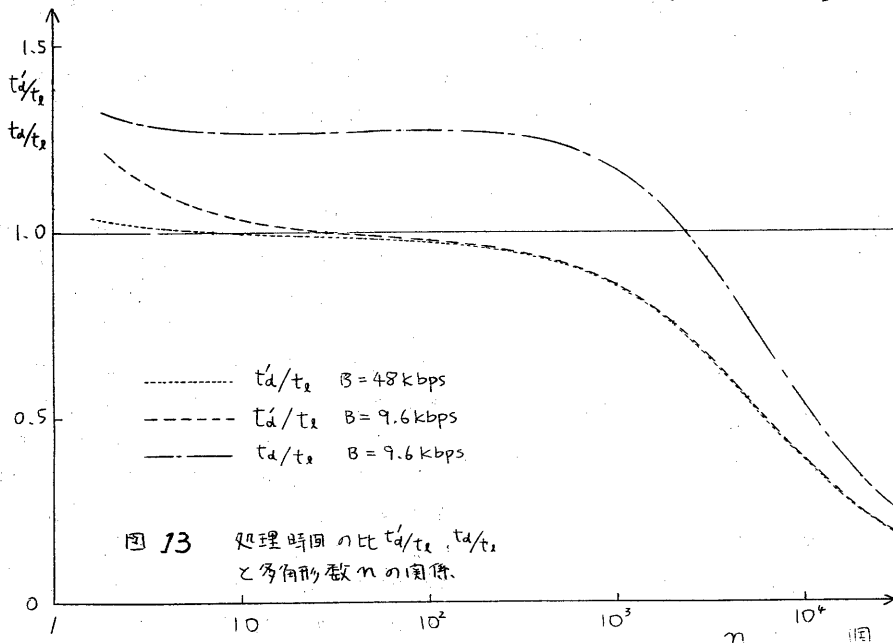


図 13 処理時間の比  $t_d'/t_e, t_a/t_e$  と多角形数  $n$  の関係

こで、ソースプログラムをプリプロセッサに通し、利用者記述のホスト側プログラムの通信制御プログラムを自動生成することが考えられる。

## 6. 検討事項

隠面消去は面の数の2乗に比例する処理なので、面をクラスタリングして部分ごとに隠面消去するのが有効である。クラスタ間の優先度の決定にはトポロジカルな性質を利用した方法があり、このためには上位に幾何モデリングシステムのようなプログラムが必要である。

幾何モデリングシステム(GMS)は計算機内で3次元物体を扱うシステムである。GMSはその内部表現としてプリミティブによる表現(CSG)を中心に、各応用によって種々の表現形式を持つものが、今後、有効であると考えられている。表示のためには、境界表現(B-reps.)が適している。B-reps.から表示ルーチンに対し、頂点や線、面、クラスタリングの情報やシェーディング情報などの表示処理に有効な情報を提供できる。したがって図14のようにGMSと3DPKGをホスト側に置き、3DPKGはGMSに対して特別なインタフェースを持つことが考えられる。

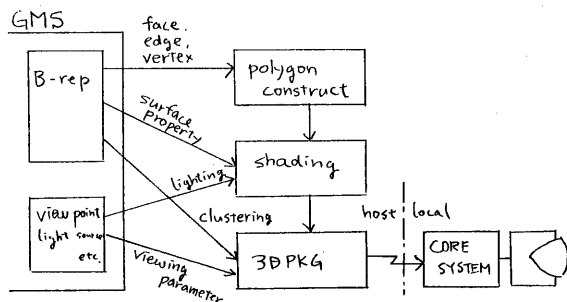


図14 GMSと表示ルーチン

## 7. 結論

本研究では分散型図形表示システムのモデルを考え、これに即して実験システムにおいて分散型図形表示システムの実装を行なった。実装システムではローカル側のグラフィックスパッケージとしてコアシステム準拠のMT-COREを実装した。ホスト側のライブラリとして3次元変換と隠面消去を行なうライブラリの3次元パッケージを実装し、応用プログラムからこれらをサブルーチンコールできるシステムを作成した。

この実験システムの作成及び評価により、①ホスト側のライブラリと応用プログラムとのインタフェースを3次元コアシステムのインタフェースと同様にすることが可能で、これにより応用プログラムのポータビリティが保存されること、②処理時間は分散型システムの方がスタンドアロン型システムより短くなること、③メモリ容量などの面や上位の幾何モデリングシステムとの関係から3次元処理をホスト側に配置することは妥当であること、等が明らかになった。

今後の課題としては、上位ルーチンとしての幾何モデリングシステムと実験システムとの関係、使いやすい利用環境の構築等が挙げられる。

## < 参考文献 >

- (1) 大金, 「分散型図形表示システム」, 昭和58年, 東京大学工学部修士論文.
- (2) 大金, 田中, 元岡, 「分散型グラフィックスシステムの構成に関する検討」, 昭和57年, 第25回情報全大, pp1385-1386.
- (3) 大金, 田中, 元岡, 「分散型グラフィックスシステムにおける3次元図形処理」, 昭和58年, 第26回情報全大, pp1431-1432
- (4) Foley, J.D., "A Tutorial on Satellite Graphics Systems," Computer, Vol 9, No. 8, pp14-21.
- (5) "Status Report of the GSPC", Computer Graphics, Vol. 13, No. 3, Aug. 1979.