

分散型グラフィックスシステムにおける

2M-5

3次元図形処理

大金 顕二 田中 英彦 元岡 達

( 東京大学 工学部 )

1. はじめに

コンピュータグラフィックスの分野に於いても、今後、複数の計算機が接続された環境での、分散処理の利点を生かしたシステムの利用がさかんにになると考えられる。本報告では、分散型グラフィックスシステムに於ける、3次元図形処理の実装例及びその評価について報告する。

2. 分散型グラフィックス実験システム

2-1. 分散型グラフィックスシステム<sup>[1]</sup>

分散型グラフィックスシステムの構成として、グラフィックスデバイスがミニコン又はマイコン程度のローカルプロセッサに接続され、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ものを考える。

ソフトウェア構成は、ローカル側にグラフィックスパッケージが存在し、アプリケーションプログラムがローカルと、ホストに分散しているものとする。単純な処理はローカルで吸収し、大規模、複雑な計算をホスト側で行なうことによって、システムの性能、効率の向上を図るものである。

ホスト側の複雑な処理はライブラリとしてサポートされており、ローカルのアプリケーションプログラムから、サブルーチンコールの形で呼び出される。ホスト側に置かれたアプリケーションプログラムも、他のライブラリと同様の呼び出し方で使用される。

2-2. 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2次元のコアシステム<sup>[2]</sup>が実装されている。分散システムの実験システムとして、東大大型計算機センタのM200/280Hと接続し、センタ側に、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。

2-3. 3次元パッケージ

センタ側の3次元変換及び、隠面消去を行なうライブラリを3次元パッケージと呼ぶ。アプリケーションプログラムと、3次元パッケージのインタフェースは、3次元のコアシステムとほぼ同様にし、ア

プリケーションプログラムのポータビリティを保つようにする。したがって、全体として、3次元処理部分をセンタ側に分散させたコアシステムとみることができる。

3. 評価・検討

3-1. 処理時間

ここでは表示する図形の複雑度を、定義する多角形(面)の数で代表させることにし、簡単な例を用いて処理時間と面の数の関係を求めた。ここで用いた例は、視点の角度をセンタ側のアプリケーションプログラムに渡すと、センタ側で3次元パッケージを用いて複数の4角錐を生成してローカルに隠面消去した結果を送り返すものである。まず、ローカルのみで実行した場合の処理時間は次のようにかかる。

$$t_l = an^2 + bn + c_l + pn \quad (1)$$

$a n^2$ は主にプライオリティソート(隠面消去)の時間、 $b n$ は面の定義(3次元変換、クリッピング)の時間、 $c_l$ はイニシャライズ等の時間、 $p n$ は、 $n$ 面の多角形を表示する時間である。

分散システムに於ける処理時間は次のようになる。

$$t_d = \frac{a}{r} n^2 + \frac{b}{r} n + c_d + \left(\frac{T}{B} + \alpha\right) n + pn \quad (2)$$

ここで、 $r$ は、ローカルとホストの計算機の処理速度の比、 $T$ は1多角形当りのデータ転送量、 $B$ は伝送速度、 $\alpha$ はI/Oのオーバーヘッドを1面当りに換算したものとす。  $t_d < t_l$ となる $n$ の範囲を求めると、 $\frac{T}{B} + \alpha + \frac{b}{r} \gg b$ のとき

$$n > \frac{\frac{T}{B} + \alpha}{a(1 - \frac{1}{r})} + \sqrt{\frac{c_1}{a(1 - \frac{1}{r})}} \quad (3)$$

ここで  $c_1 = c_d - c_l$

$\frac{T}{B} + \alpha + \frac{b}{r} \lesssim b$ のとき

$$n > \sqrt{\frac{c_1}{a(1 - \frac{1}{r})}} \quad (4)$$

③又は④ならば、 $t_d < t_l$ となる。実験システムの場合は、③である。③の第1項は1面当りの通信オーバーヘッドを、 $n^2$ の処理時間の差で割ったもの、第2項は、 $n$ に関係しない処理時間の差を $n^2$ の処理時間の差で割ったものの平方根である。 $c_1$ は主に、分散処理の場合の初期化などのための通信時間であるか

ら、 $c_1 = (c_{int}/B) + \alpha_{int}$  とかける。各定数に実測値を入れ、3種類の伝送速度に対する③及びそのときの②を求める。このとき $\alpha$ 、 $\alpha_{int}$ は、無視する。

B = 9.6Kbps  $n > 2490$   $t_d > 2770$  sec

B = 48Kbps  $n > 507$   $t_d > 478$  sec

B = 1Mbps  $n > 27$   $t_d > 24$  sec

### 3-2. 表示とデータ転送の並列処理

以上の解析から伝送速度の高い場合には、分散システムは有効であるが、伝送速度が低い場合、分散システムが有効になるのは、比較的複雑な図形でなければならないことがわかる。これは、通信オーバーヘッドがホスト(センタ)の高速処理による処理時間短縮を上回るためである。このとき、全体の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、データ転送と面の表示を並列に行なうことにより、処理時間の短縮が期待できる。実験システムでは、9600bpsの場合、1個の多角形の情報の転送には、システムのオーバーヘッドを無視すると0.2秒程度かかり、1個の多角形の表示には約0.9秒かかる。したがって、表示の間に次の多角形の情報を転送すれば、転送時間はほとんど無視できることになる。実際は通信に伴うシステムのオーバーヘッドがあるため、多角形1個の表示時間より通信に必要な処理時間のほうが大きくなる可能性がある。この場合、適当な数の多角形をブロッキングして、転送することにより、システムのオーバーヘッドを軽減することが考えられる。

表示とデータ転送を並列化した場合③は、

$$n > \frac{k(\frac{T}{B} + \alpha)}{a(1 - \frac{1}{r})} + \sqrt{\frac{c_1}{a(1 - \frac{1}{r})}} \quad (3)$$

$$\frac{T}{B} + \alpha \leq p \quad \text{のとき} \quad k = 0$$

$$\frac{T}{B} + \alpha > p \quad \text{のとき} \quad k = 1 - \frac{p}{\frac{T}{B} + \alpha}$$

とかける。kが小さいと考えると、ほとんどの場合が④とみなすことができる。この場合、初期化に伴う転送オーバーヘッド( $c_1$ )を高速処理による時間短縮( $a(1 - \frac{1}{r})$ )が補う程度に図形が複雑な場合、分散システムが有効になる。④を各伝送速度で計算すると、

B = 9.6Kbps  $n > 38$   $t_d > 34$  sec

B = 48Kbps  $n > 17$   $t_d > 15$  sec

B = 1Mbps  $n > 4$   $t_d > 3.6$  sec

このとき、 $b \ll p$ 、 $1 \ll r$  とすればnの大きいとき、 $t_d$ と $t_l$ の比は次のようになる。

$$\frac{t_d}{t_l} = \frac{1}{r} + \frac{1}{\frac{a}{p}n+1} \quad (4)$$

$n \rightarrow \infty$ のとき処理時間比は両計算機の色速度比になる。 $1/r$ への近づき方は $a/p$ の値によって決まる。

### 3-3. 検討事項

以上の処理例では、すべての多角形(面)を一元的に扱い、環境内の全部の面を一回の隠面消去の対象にしている。しかし、実際の環境では、面がいくつかのグループに集まる場合が多い。処理時間が面の数の2乗に比例するため、すべての面を一元的に扱うのは得策ではなく、環境内の面をクラスタリングして、各クラスタ内の隠面消去を行ない、次にクラスタ間の表示優先順位の決定を行なう方法が有効である。このとき、クラスタをコアシステムのセグメントに対応させ、クラスタ間の優先順位をセグメントの優先順位とすることが考えられる。クラスタ間の優先順位決定には、Schumackerのアルゴリズム<sup>[3]</sup>のように環境のトポロジカルな性質を利用した方式が考えられる。

コアシステムでは物体を意識しないため、物体の反視点向面も隠面消去の対象に入れてしまっている。反視点向面の削除は、視点の位置と面の向きだけで簡単に計算でき、対象面を約半数にすることができるため何らかの方法で行なうことが必要である。

クラスタリングや反視点向面の削除を統一的に扱うためには、コアシステムとホスト側のGeometric Modeling System (GMS)の関係を検討しなければならない。問題となるのはGMSの内部表現、例えばB-repからコアシステムを用いて表示を生成する方式、B-rep内での属性等のデータ形式、視点や投影面の指定方法などである。また、特に分散システムの場合、問題となるのはローカルにおけるグラフィック入力とGMSの関係である。

### 4. おわりに

本報告では、分散型グラフィックスシステムに於ける3次元処理について、実装システムとその評価を述べた。データ転送と表示の並列化により比較的低い伝送速度でも分散システムは有効となる。今後の検討課題としては、GMSと分散システムの関係が挙げられる。

#### <参考文献>

- [1] 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57年、10月、第25回情報処学会理全国大会予稿集、pp. 1385-1386.
- [2] "Status report of the GSPC", Computer Graphics, vol.13, no. Aug. 1979.
- [3] I. E. Sutherland, R. F. Sproull, R. A. Schumacker, "A Characterization of Ten Hidden-Surface Algorithms", Computing Surveys, vol.6, no. 1, March, 1974.