

2プロセッサ構成によるグラフィックパイプラインの最適化*

2J-10

佐藤健哉† 高橋 弘‡

† 住友電気工業株式会社 情報電子研究所§

‡ 住友電工システムエンジニアリング株式会社¶

1 はじめに

PHIGS+(Programmer's Hierarchical Interactive Graphics System) [1] [2] に代表されるように、3次元のグラフィックスの処理は、複合(ローカル、グローバル)モデリング変換、モデリングクリップ、ビューオリエンテーション変換、光源シェーディング計算、ビューマッピング変換、ビュークリッピング、ワークステーション変換などであり、図形は、これらの処理を行った後にはじめて表示される。通常、これらの計算を忠実にしようと計算量が大きくなる。ここで用いた手法は、2つのプロセッサをFIFOで接続し、かつプロセッサ間に共有メモリを配置する構成を採るシステムにおいて、処理に必要なデータを共有メモリで管理し、実行すべき演算に応じて処理を各プロセッサに分散させ、効率的に計算を行う(Optimally Distributed Loading)ものである。

2 システム構成

2.1 ハードウェア

システムのハードウェア構成を図1に示す。ホストプロセッサ(HP)、ジオメトリパイプラインプロセッサ(GPP)、レンダリングプロセッサ(RP)で構成されている。

GPPは、2つのプロセッサ(X1、X2)でパイプラインを構成しており、HP、X1、X2、RPの間はFIFOを用いてデータを受渡しするようになっている。また、データを管理するためにX1、X2どちらからでもアクセスできる共有メモリを持っている。また、この共有メモリはHPからもアクセス可能となっている。

2.2 ソフトウェア

HPにおいては、ユーザからの入力を基にプリミティブの座標および色のデータを作成し、これをGPPに送る。PHIGSの場合のストラクチャトラバースは、HPで行なっている。GPPでは、座標変換、クリッピング、光源によるプリミティブの色計算を行う。これ以外にも自由曲線、曲面の生成やピックの処理もGPPで行なっている。ここで作成されたデータをRPに送り、RPが描画を行う。

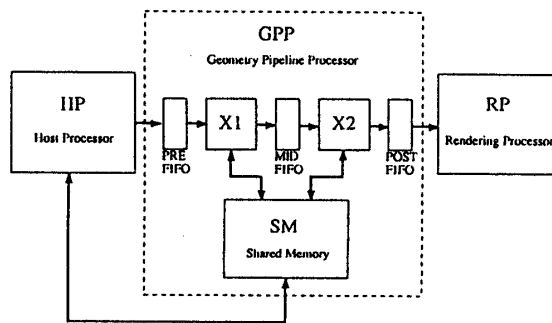


図1: システム構成

GPPにおいては、計算を行うべき状態やアトリビュートに応じて、2つのプロセッサの負荷をなるべく均等になるように、処理をX1からX2、X2からX1といったように移動、マージを行い、GPP全体として効率良く計算を行う。座標変換のためのマトリクス、プリミティブのアトリビュートのデータは共有メモリで統一的に管理し、処理がX2からX1に移ったときも、それに付随するデータを移動させなくてもよいようになっている。

3 ジオメトリ計算のモデリング

PHIGSのレンダリングパイプラインは通常、複合モデリング変換、モデリングクリップ、ビューオリエンテーション変換、光源シェーディング計算、ビューマッピング変換、デプスキューイング、クリッピング、ウィンドウビューポート変換のように構成されている[3]。本システムでは、これらの処理を、座標変換(Vertex Xform)、モデリングクリップ(Model Clip)、ベクター変換(Normal Xform)、ベクター正規化(Normal Normalize)、ライティング(Light Compute)、ビュークリップ(View Clip)、プロジェクション(Project)、ビューポート変換(View Xform)として計算をおこなっている。実際には、これ以外にも法線ベクトルの判定、デプスキューイングなどがある。ここでは、グローシェーディッドポリゴンおよびラインを取り上げる。

3.1 グローシェーディッドポリゴン

通常のグローシェーディングの場合、2つのプロセッサにおける処理分担は、図2(Normal Gouraud-Shaded Polygon Pipeline)のように、X1は座標変換、ベクター変換、モデリングクリップ、ベクター正規化、

*The Optimal Graphic Pipeline with 2 Processors

†Kenya SATO (sato@rinfo.sumiden.co.jp),

‡Hiroshi TAKAHASHI (tako@rinfo.sumiden.co.jp)

§SUMITOMO ELECTRIC INDUSTRIES, LTD.

¶Sumitomo Electric Systems & Engineering Co.,Ltd.

X2は、ライティング、プロジェクション、ビュークリップ、ビューポート変換を行う。ここで、1秒間に1座標点の処理をするとして、浮動小数点の演算量で評価すると、座標変換 24FLOPS、クリッピング 6FLOPS、ベクター変換 15FLOPS、ベクター正規化 24FLOPS、ライティング 28FLOPS、プロジェクション 11FLOPS、ビューポート変換 9FLOPSとして試算する [4]。この場合、X1は 69FLOPS、X2は 54FLOPSとなり、全体としてのスループットは 69FLOPS となる。しかし、例えば投影変換において透視投影ではなく平行投影を指定する場合、図 3 (Optimized Gouraud-Shaded Polygon Pipeline) のようにプロジェクションの処理を X1 の座標変換にマージし、それに伴って計算量の減少した X2 において、X1 で行なっていたベクター正規化の処理を行なうと、X1 が 45FLOPS、X2 が 67FLOPS となり、スループットは 67FLOPS となる。この場合、X2 のプロジェクション計算のためのマトリクスデータを X1 へ、ベクター正規化のためのデータを X1 から X2 へ、転送する必要が生じるが、ここではこれらのデータを共有メモリで管理しているために、X1 のデータを X2 に転送したり、あるいは再計算、X1、X2 の両プロセッサでデータを管理する必要がない。この共有メモリにより、処理を実行するプロセッサを容易に変更できるようになっている。

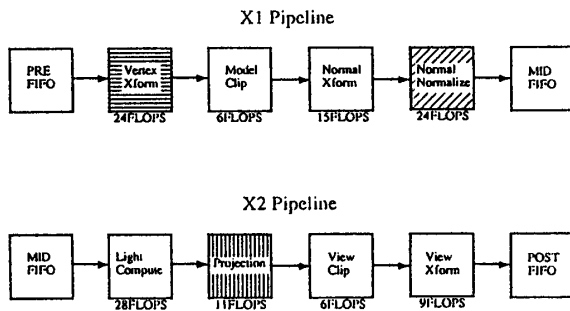


図 2: Normal Gouraud Polygon Pipeline

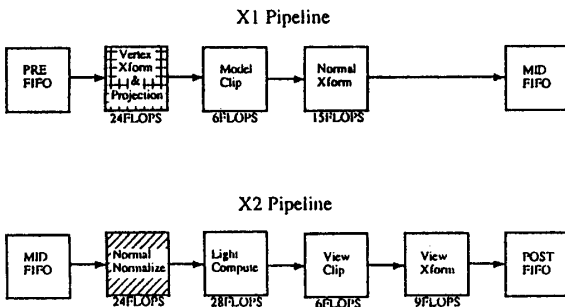


図 3: Optimized Gouraud Polygon Pipeline

3.2 ライン

単一色のラインの場合、図 2 のパイプラインをそのまま適応したとすると、X1 が 30FLOPS、X2 が 26FLOPS でスループットとしては、30FLOPS となる。このパイプラインを図 4 (Optimized Line Pipeline) のように変更すると、ポリゴンの場合、X2 で行なっていたプロジェクションの計算を、X1 において座標変換のマトリクスと合成して処理を行なう。クリッピングの計算はすべて X2 で行なう。処理量は、X1 が 24FLOPS、X2 が 15FLOPS となる。処理性能が 1.25 倍になるばかりでなく、例えば、クリップ領域として任意の平面を新たに 1 つ定義をしたとしても (6FLOPS)、X2 の処理量が 21FLOPS となるが、全体としてのスループットが低下することはない。

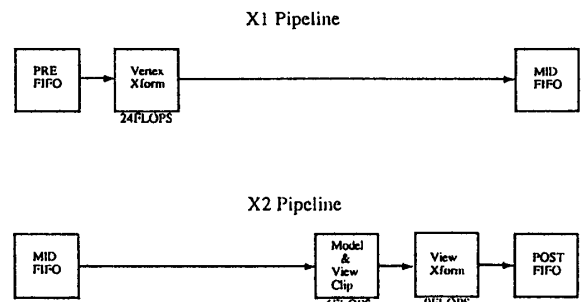


図 4: Optimized Line Pipeline

4 おわりに

本稿で述べたようなグラフィックスパイプラインにおいて、通常ボトルネックとなるジオメトリの計算を 2 つのプロセッサにおいて最適化を行なうことで、グローシェーデッドポリゴン、ラインが高速に処理でき、処理のためのデータ、プリミティブの属性を統一的に管理できる。本稿の例以外にも、フォンシェーディングや様々なプリミティブの場合に適用することができる。現在、このグラフィックスパイプラインの最適化に基づいた製品を開発中である。

参考文献

- [1] *Programmer's Hierarchical Interactive Graphics System (PHIGS)*, Draft International Standard ISO DIS9592-1, International Standard Organization, October 1987.
- [2] Andries van Dam, *PHIGS+ Functional Description*, Revision 3.0 February 5, 1988.
- [3] Randy J. Rost, *PEX Introduction and Overview* PEX Version 3.20, October 10, 1988.
- [4] Kurt Akeley, Tom Jermoluk, "High-Performance Polygon Rendering" Proceeding of SIGGRAPH'88 (Atlanta, August 1-5, 1988). In *Computer Graphics* 22, 4 (August 1988)