

CGにおける並列処理の一形態

久良知国雄 大西啓修 横木順一 宮崎正剛

三菱プレジジョン

ハイビジョン対応のCG画像生成用高速並列処理マシン(MAGG)におけるZバッファレンダラの構築について述べる。MAGGは、浮動小数点演算機構内蔵の32ビットマイクロプロセッサT800を最大で119個実装し、共有メモリ方式と疎結合マルチプロセッサ方式を階層的に組み合わせたシステムである。Zバッファレンダラは、並列処理言語 Occamを用いて開発し、処理ステップのパイプライン化とオブジェクト空間及びイメージ空間での並列化の手法を組み込んでいる。フォンシェーディングと各種マッピングを施し、アンチエイリアシングを行った高品位のハイビジョンCG画像をスーパーミニコンの30倍から60倍の速度で高速に生成する。

A PARALLEL PROCESSING FOR COMPUTER GRAPHICS IMAGE GENERATION

Kunio Kurachi Hironobu Onisi Junichi Abeki Seigo Miyazaki

Mitsubishi Precision Co., Ltd.

345 Kamimachiya, Kamakura City 247, Japan

The Z-Buffer rendering algorithm for computer graphics image generation applied to the high performance parallel processing system "MAGG" is discussed. The MAGG contains 119(maximum) 32bit micro-processor chips (T800 includes FPP), using hardware architecture based on a hierachical combination of loosely coupled multiprocessor with shared memory approach. The Z-Buffer renderer written in OCCAM language uses a technique of pipelining and parallelizing in object-space and image-space, resulting 30 to 60 times faster performance comparing to the conventional super-minicomputers was obtained for rendering of HDTV CG image by the Phong-sharding algorithm with various types of mapping and anti-aliasing.

1. はじめに

NHK殿との共同開発による三菱高速画像生成装置(MAGG: Mitsubishi Advanced image Generator)は、ハイビジョンに対応した並列処理型の高速レンダリング・マシンである。

本システムの特徴として以下の点があげられる。

(1) 階層化された並列処理による高速化

ハイビジョンのCG画像生成は、NTSCの約6倍という画素数のために、より多大な計算量を必要とする。MAGGは、浮動小数点演算機構内蔵の32ビットマイクロプロセッサT800[20MHz]を標準で85個、最大で119個実装し、共有メモリ方式と疎結合マルチプロセッサ方式を組み合わせた並列処理方式により、CG画像生成を高速に行う。

(2) レンダリングアルゴリズムのファームウェア化

レンダラは、ファームウェアとしてすべてプログラムにより実現され、拡張性や融通性を確保している。つまり、MAGGは、様々なアルゴリズムに対応していける画像生成のプラットフォームとして並列処理環境を提供しており、様々な種類のレンダラがインプリメント可能となっている。

(3) アニメーション制作に適したデータ構造

グラフィックスデータは、MAGGの中に、シーン、カメラ、物理特性、光源、階層構造をもつオブジェクトデータといった、抽象化された概念データとして蓄積され、操作できる。

(4) 実用性の高いシステム構成

様々な入出力デバイスが接続可能であり、VTRやデバイスのコントロールをMAGGから直接行うことで、ハイビジョンやNTSC規格のアニメーションの自動コマ取り収録を画像生成とVTR収録で並行して行うことができるなど、アニメーションの制作現場で要求される実用性を備えている。

本稿では、まず、MAGGのハードウェアアーキテクチャの並列処理環境を記述し、現在、標準で搭載されているZバッファレンダ

ラの機能と並列処理の形態について報告する。

2. MAGGのハードウェアアーキテクチャ

2.1 システム構成

図1のシステム構成例で示される様に、MAGGはホストコンピュータに接続される高性能グラフィックスエンジンとして、バックエンドプロセッサの形態で使用される。

2.2 ハードウェア構成

図2は、MAGGのハードウェア構成である。

描画プロセッサボードは、ハードウェアの中核となるボードであり、図3にその内部構成を示す様に、管理プロセッサ的存在の1個のFPと、作業プロセッサ的存在の16個のNPの合計17個のT800が実装され、最高7枚まで装着可能である。

2.3 並列処理環境

MAGGのハードウェアアーキテクチャが提供する並列処理の環境は、以下の様になる。

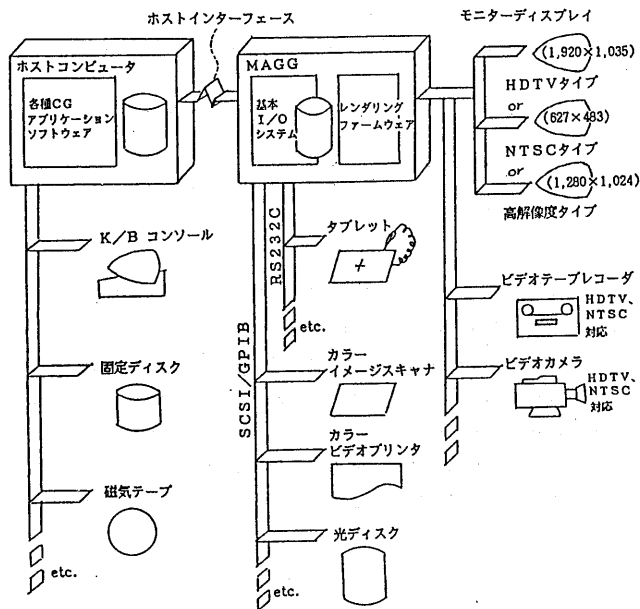


図1 システム構成例

(1) 共有メモリ方式

制御プロセッサボードの68020と各描画プロセッサボードのFPであるT800のプロセッサ群は、描画バスを経由してそれぞれお互いのボード上の共有メモリをアクセスできることから、共有メモリ方式の並列処理を行える。プロセッサ間の同期制御やデータ排他制御の手段としては、各ボード間を結ぶシリアルリンクによるパケット通信や、バスロックによる排他制御機構を使用できる。

(2) 疎結合マルチプロセッサ方式

描画プロセッサボード内のT800のプロセッサ群は、それぞれ独立したローカルメモリをもつ上、T800の

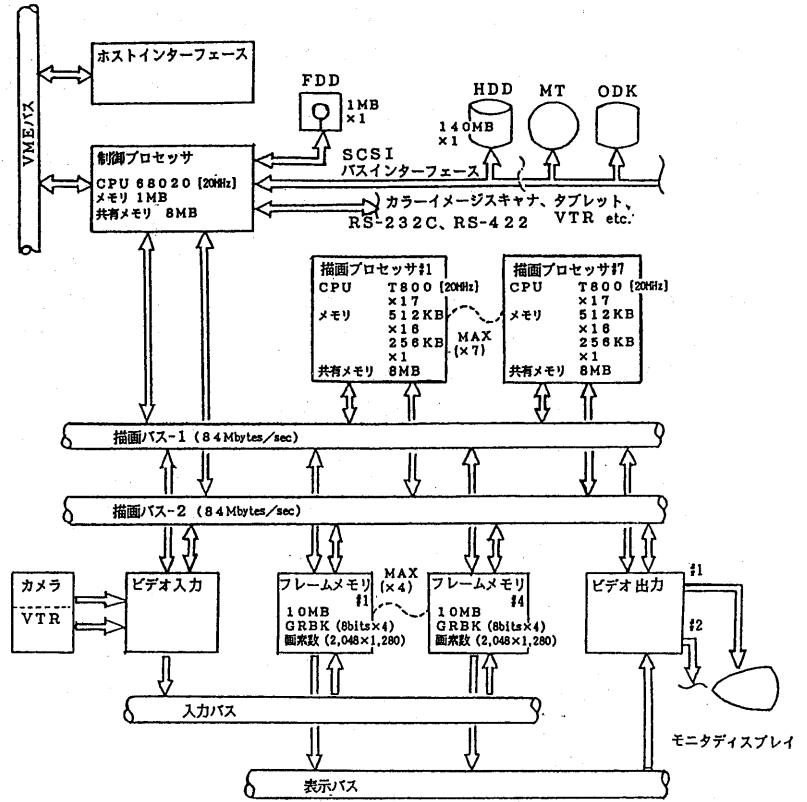


図2 ハードウェア構成

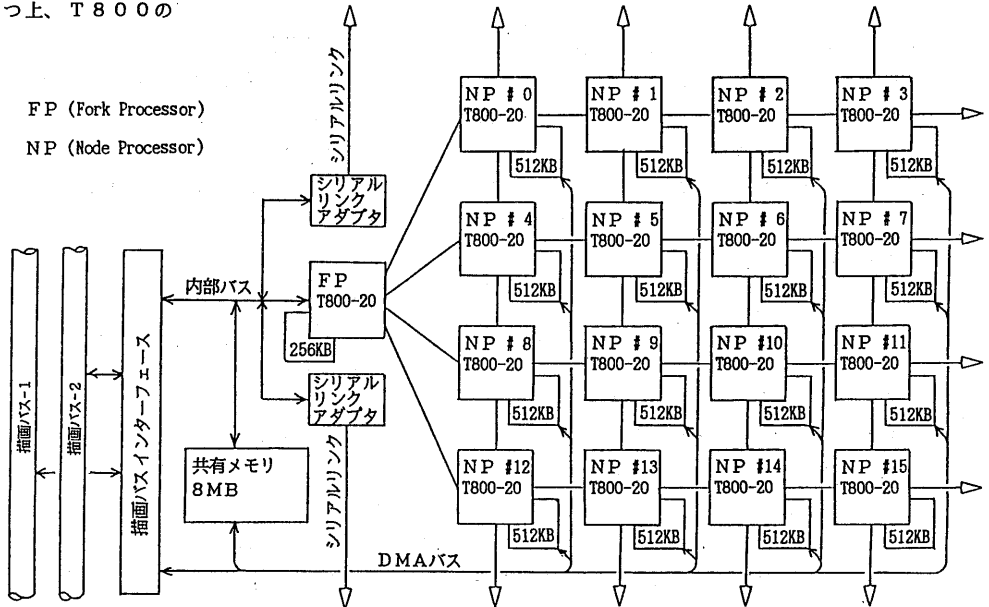


図3 描画プロセッサボードの内部構成

双方向リンク結合により、ボード内では格子結合、さらにボード間リンク結合によれば、円柱、メビウス、トーラス結合など様々なトポロジーが組めることから、疎結合マルチプロセッサ方式の並列処理を行える。プロセッサ間のメッセージの交換手段は、リンクによるバケット通信を使用できる。

(3) 階層化された並列処理環境

共有メモリをアクセスできる描画プロセッサボードのFPが、同時にNPとのネットワークを形成しているため、共有メモリ方式と疎結合マルチプロセッサ方式の階層化された並列処理環境が実現されている。直接アクセスができプロセッサ間通信が容易な共有メモリ方式と、メモリ競合の無い疎結合マルチプロセッサ方式のそれぞれの長所を生かした形態で処理を割り当て、システムを構築できる。

(4) リンク通信とDMA転送

T800やシリアルリンクアダプタのリンク経由で、同期を必要とする小量データの通信を行えるとともに、DMA転送により、NPのローカルメモリとボード内外の共有メモリやフレームメモリとの双方向データ転送や、ボード内共有メモリから全NPのローカルメモリに対しての同報転送が、大容量データについて高速に実行できる。

(5) 並列処理向マイクロプロセッサT800

T800は、浮動小数点演算機構を内蔵する他、4チャンネルの高速リンク[20Mbps]や高速タスクスイッチングなどマルチプロセッサシステムを支援する機構をもち、並列処理言語 Occamによる効率の良いプログラミングができる。

3. Zバッファレンダラ

3.1 レンダリング機能

Zバッファレンダラは、ツリー状の階層化オブジェクトへ幾何属性・表示属性を柔軟に間接指定可能なMAGGのグラフィックスデータをもとに、以下の様な特徴となる機能を備えている。

(1) Zバッファ

ダイナミックレンジの広い32ビット浮動小数点Zバッファを用い、隠面判定を行っている。

(2) 光源

光源の種類として、環境光、点光源、平行光線、線状光源、円柱光源、スポット光源をサポートし、光源色、強度、方向などを任意に指定できる。また、光源のグループ化や、オブジェクト毎に有効な光源の設定が可能である。

(3) 物理特性と輝度計算

色要素毎に指定可能な、環境光反射係数、拡散反射係数、鏡面反射係数、フオン指数と、透過係数、金属特性係数、減衰率等にもとづいた輝度計算により、質感、色を表現する。

(4) シェーディング

シェーディングタイプとして、フラット、グーロー、フオンをサポートし、オブジェクト独立に指定できる。

(5) アンチエイリアシング

Aバッファ法によるアンチエイリアシングおよび透明物体表示を行う。

(6) マッピング

ミップマップ形式によるマッピングデータをもとに、テクスチャ、バンプ、リフレクションの各マッピングを単独または複合してオブジェクトに処理できる。

(7) バウンディングボックス

ある範囲内に存在するオブジェクトをグループとしてバウンディングボックスを定義し、高速クリッピング、描画順序の最適化を行う。

(8) キー値計算

GRBの輝度値の計算の他、実写映像などとの合成のためのキー値を計算できる。

(9) ガンマ補正

ディスプレイ装置のガンマ特性を考慮した補正を行う。

(10) 背景合成

フレームメモリに存在するCGや実写のイメージデータを一定のZ値をもつ背景として、その上にオブジェクトを描画できる。

3.2 ファームウェアの構成

ファームウェアは、制御プロセッサボード上の制御プロセッサ部と描画プロセッサボード上の描画プロセッサ部の大きく2つに分けられ、制御プロセッサ部(IOP)では、リアルタイムマルチタスクモニタの下で、描画プロセッサファームウェアのロード、ホスト

コンピュータとの通信、グラフィックスデータの管理、コマンド制御、描画制御、VTR制御などの処理が行われ、描画プロセッサ部において実際の各種レンダリングアルゴリズムに基づく画像生成処理が実行される。

Zバッファレンダラとしての描画プロセッサ部は、1枚の描画プロセッサボードで構成するジオメトリック処理部(GP)と残りのボードで構成するレンダリング処理部(RP)に分けられ、GPはIOPを、RPはGPをそれぞれマスタとしたスレーブ動作を行う。

Zバッファレンダラの描画プロセッサ部は、すべて並列処理言語 Occamによって開発した。

3.3 処理概要と並列化

図4は、Zバッファレンダラの処理概要を示したものである。

MAGGの階層化並列処理環境へのデータ配置とタスク集合の割り当ては、Zバッファレンダラの場合、取扱グラフィックデータの量的能力を優先し、そのアルゴリズムの並列性検出をもとに並列処理効果を向上させる次の3通りの手法を基本としている。

- (1) 処理ステップのパイプライン化
IOP、GP、RPの各ステージで、マクロなパイプライン化を行う。
IOP-GP (RP)間では、フラグ制御によるグラフィックスデータの更新と描画処理が、また、トークンによるリソース管理をもとにコマ撮りと描画処理とが、それぞれオーバーラップして行なわれる。
GP-RP間では、オブジェクト単位に、

ジオメトリック処理からレンダリング処理への描画処理のスループットに最も影響するパイプライン処理が、共有メモリを介してシリアルリンクのバケット通信による同期制御を用いて行われる。

(2) オブジェクト空間での並列化

ジオメトリック処理が行われるGPでは、各NPにオブジェクト単位の座標変換を割り当てて、並列処理を行う。IOPによって展開されているグラフィックスデータを、ボード内の各NPの稼働状況を管理しているFPが、次々とビジーでないNPのローカルメモリへ、別々のオブジェクトデータを転送し座標変換を依頼する。

したがって、全NPで最大16個のオブジェクトについて並列に処理が行われる。変換結果は処理の終了時点でNPのローカルメモリからGPの共有メモリへ転送される。

(3) イメージ空間での並列化

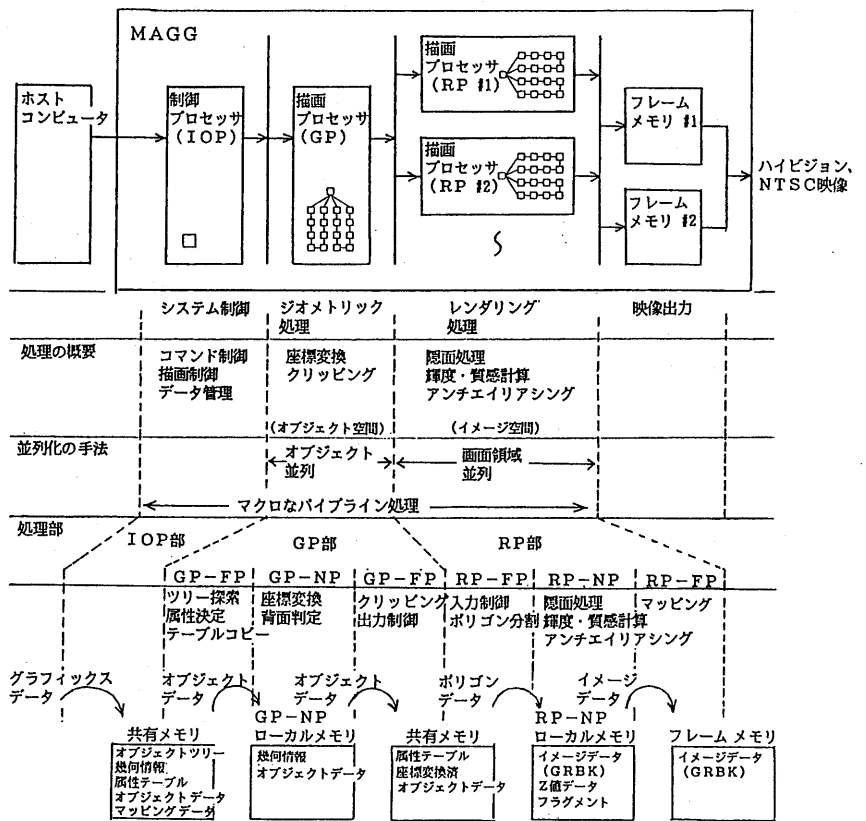


図4 Zバッファレンダラの処理概要

レンダリング処理が行われるRPでは、GPの指示に基づき、ボード毎に矩形画面領域と各NP毎に等間隔スキャンラインを割り当てられ、負荷分割による並列処理を行う。GPによる座標変換後のグラフィックスデータは、FPが領域判定した、ボードでの有効ポリゴンデータを全NPのローカルメモリへ共通にデータ転送し、それぞれの担当領域についての描画処理を依頼する。標準構成で4枚のRPは、ボード毎に、画面の縦横それぞれを2等分した4分の1の領域が与えられ、その中の16本おきのスキャンラインが、各NPに割り当てられる。描画処理の終了時点で、作成されたNPのローカルメモリのイメージデータはフレームメモリに転送される。

3. 4 実現方式の評価

並列処理の効率は、データの授受が特に問題となる。

Zバッファレンダラでは、次にあげる2点について、それぞれ2通りずつの方式を考え、実際の評価は、それぞれの方式を採用して実現した3つのモデルI、II、IIIについて行った。

(1) GP-RP間のオブジェクト授受の制御

一般に、共有メモリ方式では、プロセッサが増えていくと共有メモリへのアクセスが集中し、それがボトルネックとなって、性能低下を起こす。

3. 3 (1) のパイプライン処理に関しては、GP-RP間のオブジェクト授受の制御について、次のaとbの方式を考えた。

a. メッセージバスによる方式 (モデルI、II)

GPからシリアルリンクによるケット通信で1つ1つのオブジェクトの共有メモリ上の幾何データ位置を指示し、その応答をRPから同じくシリアルリンクのケット通信で毎回返して同期をとる方式。インターフェースにおけるメモリの競合は起きないが、シリアルリンクの通信路の使用頻度が高い。

b. 共有メモリ上のセマフォによる方式 (モデルIII)

GPからオブジェクトの共有メモリ上の幾何データ位置を同じく共有メモリ上の間接参照テーブルに書き入れていき、シリアルリンクによるケット通信では同期をとるセマフォとテーブルの位置及び参照開始のみを指示する方式。インターフェースにおけるメモリの競合は起き

やすいが、シリアルリンクの通信路の使用頻度は低い。

(2) RPのFPとNP間のポリゴンデータ転送

一般に、疎結合マルチプロセッサ方式では、プロセッサ間のデータ転送を転送幅の狭い通信ポートによって行うのみでなく、直接接続されていないプロセッサ間でのデータの授受を他のプロセッサを経由して行うため通信のためのコストがかかるが、先に述べたように、T800では、マルチプロセッサシステムを意識した比較的高速なリンクによる通信路をもっている。また、MAGGの場合は、共有メモリやフレームメモリとNPローカルメモリ間でDMA転送により大量データを高速に転送できる。ただし、DMA転送の設定には、FPとNP間のリンクによる制御ブロックの通信と同期を必要とする。

3. 3の(2)と(3)の並列化の手法に関する特徴は、GP、RP、どちらのNP群においてもNP相互間のデータ依存関係が無い反面、NPローカルメモリと共有メモリ(フレームメモリ)間でのデータ転送量が膨大になることである。3. 3(2)のGPで異なるオブジェクトを単一NPのローカルメモリ毎にデータ転送する場合と、3. 3(3)のRPでイメージデータを処理の終了時に非同期に各NPローカルメモリからフレームメモリの別領域へ転送する場合は、1対1の大量データ転送という点で、DMA転送の効果はリンク転送に対して明らかである。しかし、3. 3(3)のRPでポリゴンデータを各NPローカルメモリへの転送する場合は、オブジェクト単位のデータやイメージデータに比べ、データ量の単位が小さいため、1対1のDMA転送で実現すると16倍の転送回数が必要となり、同一ポリゴンデータをボード内全NPへ一括転送するDMA同報転送では転送のタイミングの調整とNPの稼働率の低下の問題が発生する。

RPのFPとNP間のポリゴンデータの転送は、NPの制御を含めて次のcとdの方式を考えた。

c. リンク転送による方式 (モデルI)

ポリゴンデータの転送を比較的高速なT800のリンクによる通信経路ですべて行い、ボード内の各NPの処理時間のアンバランスによる無駄時間を短縮するために、16個のNPを4個ずつ4グループに分けて、ポリゴンデータの放

送を行い制御する方式。有効ポリゴンの判定はより有利に行われ、グループ別の転送量は減少できる。

- d. DMA同報転送による方式(モデルII、III)
 ポリゴンデータの転送をある単位でブロック化して、DMA同報転送によりボード内全NPのローカルメモリへ転送し、全NPを一括制御する方式。ブロッキング/デブロッキングのオーバーヘッドなどは発生するが、転送元と転送先でブロック単位のFIFOを設置し、NPの稼働率の低下を防いでいる。

図5はこの方式の形態を示したものである。

表1は、3つのモデルについての測定結果であり、写真1は、テストデータの測定時の表示出力である。

方式aに比べ方式bの優位な結果が出ているが、オブジェクト数が多くそのオブジェクトのポリゴン数が少ない場合に顕著である。この結果から、セマフォやテーブル参照によるメモリ競合の影響よりも、速度の遅いシリアルリンクによる通信路がボトルネックとなっていることがわかる。

また、方式cに比べ方式dの優位な結果が出ているが、これは総ポリゴン数が多い場合に顕著である。この結果から、DMA同報転送による高速な転送がデータの供給に活かされていることがわかる。

3つのモデルでは、光源数が多くなるに従って評価結果の差が縮まっていることがわかる。これは、NPの計算量が増加することでデータ到着待ち時間が総体的に減少するためであると考えられる。尚、方式cについては通信バッファのサイズ、方式dについてはブロックのサイズやFIFOの段数などの評価も行ったが、ここでは省略する。

3. 5 性能評価

並列処理システムの開発目標は、n個のプロセッサを用いて、1個の性能のn倍にできるだけ近い性能を引き出すことであるが、プロセッサ間の通信や同期が必要なために、実際にはn倍に近い性能は難しい。

プロセッサ個数を変化させての速度向上率などについては、具体的なデータによる十分な評価にまで至っていないが、実際、描画領域の変更など、イメージ空間での負荷の変化に対する効率でも、様々な条件でその結果はかなり異なっている。

Zバッファレンダラは、アルゴリズム自体、レイトレーシングなどと比較して効率の良い並列化は難しいが、本レンダラでは、Data General社のスーパーミニコンピュータMV-20000上の、ほぼ同じ機能をもつプログラムでの表示時間に対する実測値での比較で、ほぼ30倍から60倍の性能であった。特に演算負荷の重い、フォンシェーディングで光源を多数使いアンチエイリアシングを施す様なケースでよい結果を得ている。

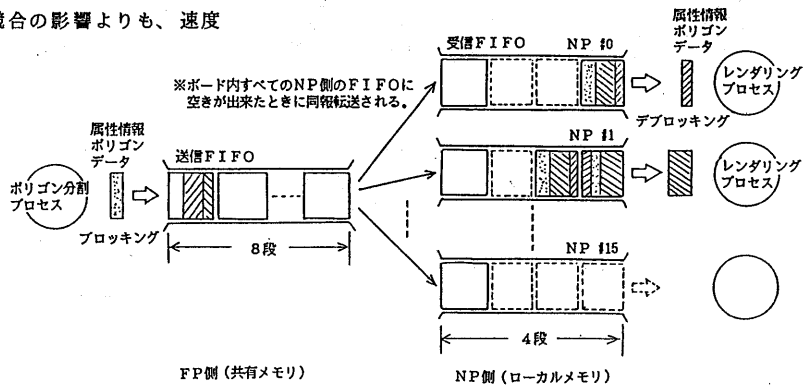


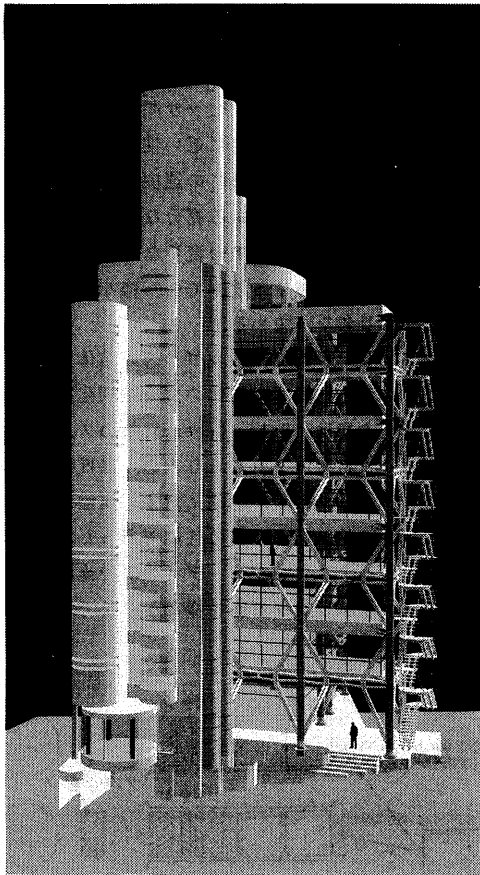
図5 DMA同報転送によるポリゴンデータの転送形態(方式d)

表1 実現モデルの測定結果

時間単位はsec

表示データ内容	オブジェクト数	総ポリゴン数	アンチエイリアス	シェーディング、その他(すべてハイビジョンサイズでレンダリング)	行#I	行#II	行#III
					aタイ*	cタイ*	dタイ*
直方体	2,000	12,000	あり	フォン、環境光1、点光源1	82.3	79.4	19.1
直方体	2,000	12,000	あり	フォン、環境光1、点光源5	83.3	80.2	42.9
球(800面体)	200	160,000	なし	フォン、環境光1、点光源5	142.5	70.2	67.5
球(5000面体)	10	50,000	あり	フォン、点光源1	37.6	23.8	23.4
球(5000面体)	10	50,000	なし	フォン、点光源1	37.1	12.0	11.6
球(2000面体)	4	8,000	あり	フォン、環境光1、点光源5	22.6	22.4	22.4
戦艦戦艦	26	32,000	あり	フォン、点光源6	24.4	21.3	20.8
K1飯倉ビル	670	35,000	あり	フォン、環境光1、平行光1(フラグメントオーバーフロー時のリトライ処理を含む)	263.4	217.9	167.1

- ※写真1
- (1) GP-RP間のオブジェクト授受の制御
 a. メッセージバスによる方式・・・I、II
 b. 共有メモリ上のセマフォによる方式・・・III
- (2) RPのFPとNP間のポリゴンデータ転送
 c. リンク転送による方式・・・I
 d. DMA同報転送による方式・・・II、III



K1飯倉ビル(設計:リチャードロジャース, パートナース, 制作:三菱地所)

写真1 テストデータの表示出力
(ハイビジョンで横向きに表示)

3. 6 レンダリングアルゴリズムでの工夫

実現上における様々な問題点のうち、ここではAバッファ法についての対処を取り上げる。

(1) 再帰プログラミングの問題

アルゴリズムの最終ステップにおける、再帰プログラムによるフラグメントのバッキング処理は、Occam では再帰記述ができないため、複数プロセスによる疑似再帰プログラミングにより実現した。

(2) 背景フラグメント発生の問題

背景ピクセル内にエッジが発生した場合に、毎回の背景フラグメントの作成は浪費となる。単一色の背景については、そのフラグメントをあらかじめ用意して共有化することにより、有限領域にとられるフラグメントの消費量を減少させた。

(3) フラグメント領域オーバーフローの問題

仮想記憶システム上のプログラミングと異なり、メモリ容量が固定であるため、フラグメント領域が不足した場合、システムとしては、画面を縦・横に2等分して4分割し、その4分の1を全体のRPで処理することを再帰的に繰り返して、各NPの担当ピクセル数を減らすことで処理を完結する様にした。

4. おわりに

以上、ハイビジョン対応のCG画像生成用高速並列処理マシンMAGGにおけるZバッファレンダラの構築について、並列化の手法と実現方式の評価を中心に記述した。

Zバッファレンダラについては、今後、マッピング処理の高速化等のチューニングを行っていくとともに、影づけやフレア表現などの機能を充実していく予定である。

また、現在、MAGGのレンダラとして、レイトレーシングのファームウェアも開発中である。

最後に、MAGG、及び、Zバッファレンダラの開発において、御指導、御尽力をいただいた皆様に深く感謝いたします。

【参考文献】

- [1] 大西啓修、精木順一、戸村和夫、吉良健二：ハイビジョンCG用高速画像生成装置の並列処理手法、信学技報、Vol.89, No.281, 1989, pp1-6
- [2] 鈴木鎮男、鈴木 繁、精木順一、宮崎正剛：ハイビジョンCG用高速画像生成装置による画像生成、信学技報、Vol.89, No.281, 1989, pp7-14
- [3] 高橋義造：計算機方式、コロナ社、1985
- [4] 尾内理起夫：Occamとトランスピュータ、共立出版、1986
- [5] Carpenter, Loren: The A-buffer, an Anti-aliased Hidden Surface Method, Proc.SIGGRAPH'84, Computer Graphics, 18, 3, July 1984, pp103-108
- [6] Williams, Lance: Pyramidal Parametrics, Proc.SIGGRAPH'83, Computer Graphics, 17, 3, July 1983, pp1-11
- [7] Rogers, David: Procedural Elements for Computer Graphics, McGraw-Hill, 1985