

MIMD 拡張構造グラフィックプロセッサ

ウォークスルー仮想現実システムに対応するレンダリング技法

池戸 恒雄

(会津大学コンピュータ理工学部ハードウェア学科コンピュータ構築学講座)

会津若松市一箕町鶴賀 0242(37)2580 / ikedo@u-aizu.ac.jp

あらまし 仮想現実空間を生み出すコンピュータグラフィックス映像のレンダリングについて新しい構造のプロセッサを提案する。対話型ウォークスルー仮想現実を創造するには膨大な映像の実時間描画処理が必須である。このためにはポリゴンレンダリング法が有効である半面、品質が劣る。このためこのプロセッサでは陰影のための反射光計算、バンプマッピングまた、ガス状形状の表示およびカメラ映像とコンピュータグラフィックとの合成回路をシングルチップ内に完全ハードウェア化すると共に、性能の拡張機能を開発した。このASICは940,000ゲートの0.3ミクロンCMOSが用いられる。本論文ではこのASICの機能および性能をハードウェア構造から解説する。

キーワード 仮想現実、グラフィックプロセッサ、シェーディング、バンプマッピング

Graphics Processor with Scalable MIMD Structure for Virtual Reality System

Tsuneo Ikedo

Computer Architecture Laboratory

The University of Aizu

Tsuruga, Ikki-machi Fukushima, Japan 965-80

Tel: 0242(37)2580, ikedo@u-aizu.ac.jp

Abstract The graphics processor with 12 embedded graphics functions was developed for creating the virtual reality image in real time. Phong and bump mapped shading, reflection and refraction mapping, gaseous object rendering, and video mapping are incorporated fully in single chipped hardware with MIMD structure. Shaded and texture mapped pixel is rendered at 3.8ns(1.2 million triangle polygon/s with 100 pixels) with hidden surface removal. The chip is fabricated with 940,000 gates standard cell, 0.3 micron CMOS. This paper describes the hardware architecture and implementation technologies of this chip.

key words virtual reality, graphics processor, Phong shading, bump mapping, Anti-aliasing

1 はじめに

人間の知識獲得の70%が視覚によるものと言われるように、映像は情報伝達メディアとしての中心を成すものである。特に仮想現実あるいはマルチメディアシステム等の応用分野においては、情報の映像化が人とコンピュータとの重要なインターフェイス手段となるだけに、映像は人間の感性に答え得るだけの十分に即応性のある、またリアリステックな品質が確保されなければならない。インタラクティブな仮想現実を生み出す映像技術の中心は幾何学モデルを中心とするコンピュータグラフィックスであるが、この要求に答えるためのグラフィックスプロセッサには専用化された極めて強力な演算処理能力が必要とされる。過去20年間において、これらを実現すべくより高速な映像生成のための数多くのプロセッサ構造が研究開発されてきた[1][2][3]。しかしながら、今だに現在の形状モデラーにおいて、我々が自然界から受け得るに等しい視覚映像をリアルタイムに得ることは困難である。またより現実に近い映像表現力をもつレイトレーシングあるいはラディオシティ法に関して従来のプロセッサアーキテクチャーにおいて実用上、リアルタイム性を得ることは困難である。現在多くのパーソナルコンピュータあるいはワークステーションに導入されている高速処理のためのCG技法にはポリゴンレンダリングが用いられる。これは面素(ポリゴン)の塗りつぶし処理により形状を表現するものであるが、リアルタイム性を得るため、直接、物体の属性や表面荒さおよび光源を考慮した陰影処理は計算コスト上行わないため映像品質が劣る。また、今日描画可能な性能(一秒間に描画できる100画素からなる3次元面素)は40-100万個程度である。仮想現実空間を作り出すための必要な条件は用途にも左右されるものの、Shading処理を含めてその10倍から100倍は必要である。このレベルの実行性能は筆者の知るかぎり今日の商用機器からは得ることが出来ない。CGの高速処理には主として2つの構造がある。一つは四則演算機能をコアとするプロセッシングエレメントを大量に且つ並列(SIMD)に用いるもので、その幾つかはポリゴンエンジンあるいはストリックアレイとして知られている[4][5]。他はグラフィックチップあるいはASICに代表されるような専用のプロセッサ構造である。後者は15年以前より、広くパーソナルコンピュータあるいはワークステーションに用いられてきた。しかしながら、性能に対するスケーラビリティ(拡張

性)がない等の問題を持ちつつ、今日に至っている。

本論文で述べるTruga001は上記の各種の問題点を解決すべく開発されたグラフィックスプロセッサである。このプロセッサの特徴は第一に高速描画性能の達成と、ポリゴンレンダリングにおいて、よりリアルな映像描画実現の反射光計算、ガス状物体のポリームレンダリングおよび陰影処理等の導入にある。このプロセッサはGVIP開発プロジェクト[6]の後継として完成したものであり、前記グラフィックス専用チップとしての構造をもつ一方、スケーラビリティとリアルタイム陰影処理を組み入れたものである。性能はベクター描画(10画素)において一秒間に400万本。ポリゴンではテキストチャーマッピング、Phongシェーディング、バンプマッピング、陰面消去および画像メモリアクセスタ임을考慮した条件下において一秒間に120万画素(100画素)の描画が可能であり、HDTVのビデオ映像をフレーム落ちなく取り込んだ場合においても、100万画素の描画性能をもつ。またスケーラビリティ構造によって、チップの並列網構造で1000万画素/秒以上の拡張が可能である。この拡張性はこのチップが特定のアプリケーションにのみ利用されるものではなく、ゲーム機器からCAD/CAMあるいは数千万ポリゴンを必要とするであろう大型アミューズメントシアターにまで対応可能とする。チップ内の回路規模はおよそ98万ゲートであり、0.3ミクロンBGAパッケージで提供される。本論文はこのTruga001の構造および性能についてその概要を述べるものである。

2. Truga 仮想現実システム

Truga 仮想現実システムは仮想視覚および聴覚研究を目的として会津大学マルチメディアセンター人工世界研究室に設置計画のものである。これは図1に示すマルチメディア統合システムの一部として位置づけられる。このシステムはスーパーコンピュータ、レンダリングプロセッサ、3Dマシブプロジェクションシステムおよび音響システムから構成される。

第一次開発システムは市販および特注製品を用い、図1のシステムは1994年7月に大学内に設置され日々平均40名の見学者を迎えている。しかし、市販品によるシステム構成のためインタラクティブアミューズメントシアターとしての仮想現実システムの性能には限界がある。この結果、第二次開発ではワークステーション部を仮想現実システムに対応する可変構築形相互接続網をもつスーパーコンピュータに置き換えるため、

ジオメトリおよびサウンドプロセッシング用超並列処理プロセッサPE5クラスターエレメントと、数千万ポリゴン描画のためのレンダリングプロセッサTruga001の2つのプロセッサの開発を中心に進められている。PE5は5つのプロセッシングエレメントを内蔵するシングルチップASICである。スーパーコンピュータはそれぞれのPCBに32Mbのローカルメモリを持つ4個のPE5を搭載し、これを6枚用いて構成する。よって120個のプロセッシングエレメント（アプリケーション専用I/Oを含む）から成り、約60Gflopsのハードウェア最大性能を持つ。図2に示すように、レンダリングシステムは48個のTruga001プロセッサを用い、それぞれ28個のプロセッサが左右の3D視差映像をそれぞれ生成し、秒間性能2千万ポリゴン（100画素）を目標とする。スクリーンシステムはすでに設置されているがこれは1.0m x 3.0mのサイズを持ち、それぞれ40インチ48個の120Hz液晶シャッター内蔵のプロジェクトから成る。

と25個のスピーカに増設する。この仮想空間は17m²の特別室に設けられている。本論文ではレンダリングプロセッサTruga001の機能および性能について述べることにし、スーパーコンピュータは別の機会に取り上げることとする。

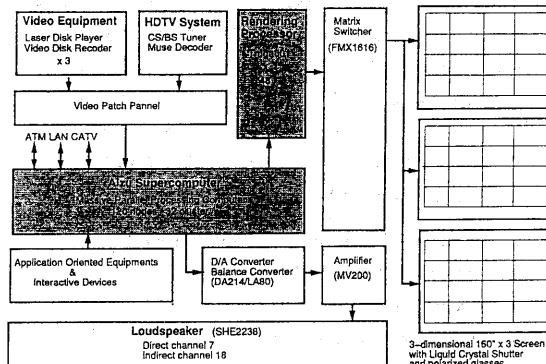


図2 Virtual Reality システム

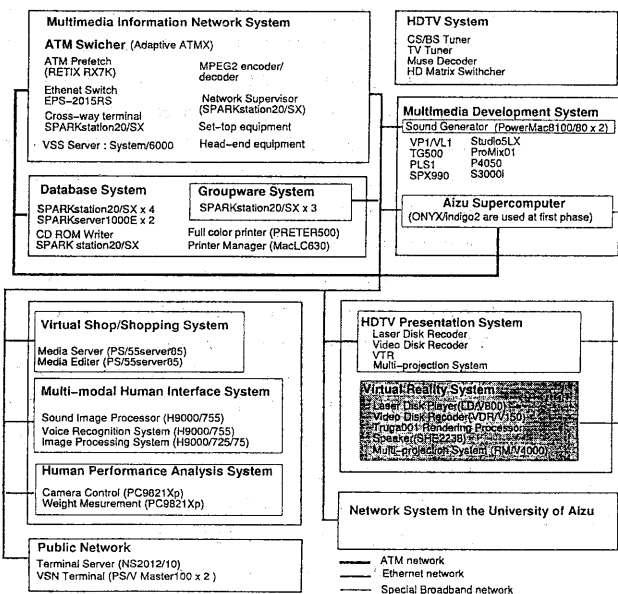


図1 Aizu Multimedia システム

このスクリーンは3つに分割されており、150度の視野角を与える。最大解像度は10,000 x 3,500である。仮想音響システムは第一段階では17個（補助低音専用スピーカを含む）のスピーカを半球状に配置しているが、第二段階では直接音7チャンネル

3 Truga001 Graphics Processor

Truga001は実時間の仮想現実空間を生成するグラフィックスプロセッサとして開発されたものである。このプロセッサがもつ主要な描画機能には、光線反射による陰影処理を含むポリゴンレンダリング、ガス状物体生成回路、またCG映像とビデオカメラ映像との合成回路等が含まれる。特に仮想現実空間をより自然体にする極めて有力な手法として、ビデオカメラ映像に対する実時間でのマッピングもチップ内部の専用プロセッサとして内蔵される。Truga001は単機能的なプロセッサ、あるいはストアードプログラム構造をもつプロセッサではなく、文字、ベクター、ポリゴン、ビデオカメラ映像、およびガス状物体の生成とそれらに対するシェーディング、画像

処理および合成等の多種の普遍的なグラフィックス機能をそれぞれ独立した複数のハードウェアモジュールによって実現した汎用的グラフィックプロセッサである。このプロセッサの基本的な設計概念は、例えばシェーディング、テキストチャーマッピング、陰面消去等の複合するグラフィックス機能が並列処理する場合においても描画性能を低下させないことをその特徴とし

ている。性能の低下に関しては唯一、互いに独立した（ポリゴンレンダリング、BitBlt、TV capturing等の多様なデータ処理）処理が同時発生した場合、画像メモリとのバスバンド幅がその性能の上限を決定することがある。しかし内部処理に関するプロセスに関してはそれぞれ独立しており、互いに影響を受けることを最小限避けている。これら並列処理を可能にするために、チップ内部には13種のプロセッサと7組の特殊回路を内蔵する。またこれら12種のプロセッサのうち、RISCCoreおよびフォントプロセッサを除いては、ハードウェア回路によって構成されている。これらプロセッサおよび回路は下記となる。

- * 32-Bit RISC Processor with 75Mflop FPP
- * Rendering(Outline Filling) Processor
- * Shading Processor
- * Phong Shading(10-light source)Processor
- * Texture (Video) Mapping Processor
- * Bump Mapping Processor
- * Reflection/Refraction Mapping Processor
- * Gaseous Object Renderer
- * BitBlt Processor
- * Polygon Anti-alias Processor
- * Shadow-casting Processor
- * Hidden Surface Removal Processor
- * Outline Font Generator
- * Reconfigurable Multi-dimensional Pixel Cache
- * HDTV Video Capture Controller
- * Arithmetic and Logic Operation Unit
- * Program Frame Buffer Controller
- * Tripple 256 x 8-bit Video Refresh Register
- * NTSC Video Output Controller
- * Cursor Generator

Truga001の全体内部構成を図3に示す。図3において回路構成は大別して7つの機能に分類され、これらはパイプラインおよび並列構造で互いに接続されている。これらには、浮動小数点演算プロセッサを持つRISCCore、レンダリングプロセッサ、ハードウェアモジュール群、ピクセルキャッシュ、Zキャッシュ、画素演算ユニット、および画像メモリ制御回路系が含まれる。RISCCoreおよび他のすべてのプロセッサはそれぞれ75MHzおよび65MHzで動作する。図3において、通常RISCCoreはチップ全体のマスタープロセッサとしての役割を担う。グラフィックスハードウェアモジュール群は更に9個のプロセッサから構成され、それぞ

れはRISCCoreおよびRendering Processorと並列あるいはパイプラインで接続される。RISCCoreはこれらプロセッサに対して命令およびデータを与える。それはローカルプログラミングによって定義されたものであったり、システムプロセッサからのものである。また各プロセッサは互いに独立して（但し、Shading Processor、テクスチャマッピング回路および陰面消去プロセッサは互いに並列に、同期して）それぞれの処理を実行する。よってこれらはMIMD(Multiple Instruction Multiple Data)並列処理構造をもつ。Gaseous Object Rendererはプリミティブ形状の多様化に対応すべく、唯一 Writable Stored Program 構造をもつプロセッサである。これらのプロセッサで生成されたピクセルはマルチプレクサを介してピクセルキャッシュにロードされる。この結果それぞれのモジュールで同時並行してピクセルが出力される場合、ピクセルキャッシュのバス上でバスコンフリクトが発生することがある。しかし、ピクセルキャッシュは3次元構造をもっているため、プロセッサからは同時に（1クロック15ns以内）32x24ピクセルを書き込むことができる。すなわちピクセルキャッシュのバスバンド幅は5.1Gbyteに相当することから実際の動作上は、それぞれのプロセッサとキャッシュ間のバス幅自体が大きな性能障害の問題となることはない。一方、外部からのVideoデータのCapturingに関してはVideo Controllerがビデオカメラ映像専用のキャッシュ（ラインバッファ）をもっているため、Videoデータはピクセルキャッシュを介することなく、キャッシュ後段に接続されたFIFOにロードされる。FIFOに記憶されたピクセルは算術論理演算ユニットを通して画像メモリに書き込まれる。算術演算回路はポリゴンアウトラインのAnti-aliasあるいは透明感生成のためのColor blendingに用いられる。この際、ソース（プロセッサ）およびデスティネーション（画像メモリ）のピクセルとの非線形変換の演算も可能である。このシステムでは一つの画像メモリ内にピクセルとZ値を記憶する。チップと画像メモリとの転送はページバーストあるいは同期式DRAMではランダムカラムアクセスが用いられる。画像メモリは、モニターに表示するためのビデオ走査読みだしアクセスが逐次行われるが、このチップではシングルポートDRAMの使用を可能にするために256x8bitのトリプルシフトレジスタを内蔵する。チップは図4に示すように大別して4つの外部バスをもつ。それぞれは2組のRISCCoreバス、384bitの画像メモリバスおよびビデオリフレッシュバスである。一つのTruga 001チップは画像メモリの

24プレーンをサポートし、一つのチップで自然色表現を行う。Truga001は一組のチップが120万ポリゴンの描画性能をもつ一方で、これを並列に接続することによって制限のない拡張が可能である。これは図5に示すように、並列化したそれぞれのTruga001から出力されるビデオデータをTruga002によって2分岐法で合成して得られる。この場合それぞれのTruga001から出力されるビデオデータはピクセルだけでなく、Z値も出力され、合成は陰面消去法に基づく。図5の構成においては約1,000万ポリゴン(100ピクセル)の実行性能を得る。

よってそれぞれのモジュール内部では互いに独立且つ非同期な動作が可能となるため、スケラビリティは単純にモジュールの数に比例して増加する。また画像メモリは表示対象領域のすべてを包括する容量をもつため、システムプロセッサはそれぞれのモジュールに対して順次受信可能なものからデータをロードすればよい。この結果、システムプロセッサはモジュール毎のポリゴン分割やロードバランスを考慮する必要がない。Truga001は32bit構造のRISCプロセッサを内蔵する。仮想アドレス管理機構を内蔵していないため本文ではRISCcoreと呼ぶ。

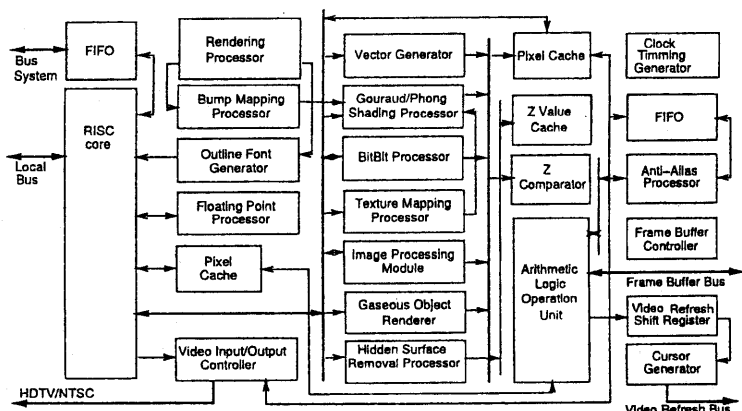


図3 Truga001 内部回路構成

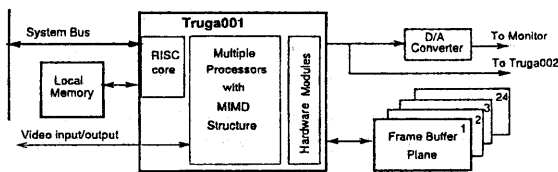


図4 全体構成

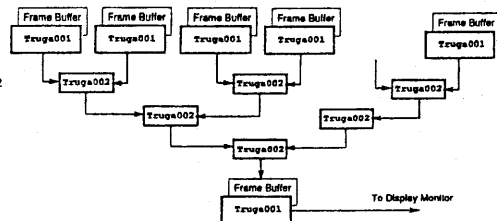


図5 並列接続構成

Truga002は2組のTruga001からそれぞれ走査みだし順に画素値とZ値を受信して画像データを合成するプロセッサである。すなわち3次元空間上にランダムに分布した画素をフィルタリングした後、透明度および減衰率(画像メモリ内に画素の一部として記憶される)をそれぞれ考慮してポリゴンとの合成を行う。それぞれのモジュールが互いに独立した画像メモリをもつ並列構造においては、この画像メモリを介することに

4 Embedded RISC-core

このプロセッサは32bit x 64wordのレジスタファイル、整数および浮動小数点演算プロセッサ(浮動小数点演算回路には32bit x 64wordのレジスタファイルを内蔵)、8および16Kbyteの命令およびデータキャッシュ、双方向FIFO、32x8bitのRISCピクセルキャッシュを内蔵する。FPPは四則演算と平方根演算が可能

である。プロセッサを取り巻くバス構造は図6に示すように3つの32bitバスからなる。システムプロセッサはFIFOを介してこのプロセッサと接続される。またローカルバスにはグラフィックデータのセグメント（ストラクチャ）バッファやローカルプログラミングのためのRAM/ROMが通常接続される。一方、グラフィックバスが設けられ、このバスを介して直接外部からハードウェアを制御する。よってRI-SCoreを使用しないシステムの構築も可能である。

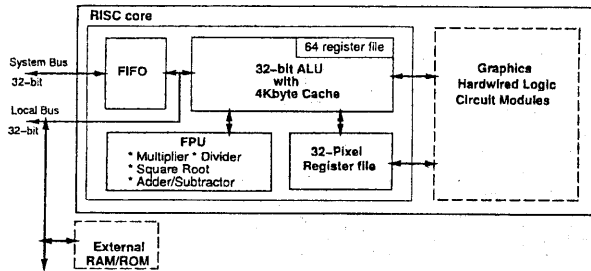


図6 Embedded RISC core 内部構成

ハードウェアモジュールは12.8bit幅のバスをもつ。これらはRISC coreとの内部グラフィックデータバス、BitBlitバス、Rendering Processor バスおよびGaseous Object Generatorバスである。このうち内部グラフィックバスには2次元配列画素の画像メモリ間の転送レイテンシを最小化するためRISCピクセルキャッシュ（ハードウェアモジュールと画像メモリ間のキャッシュとは異なる）が設けられ、32x24bitのピクセル（自然色構成の場合）を120nsで画像メモリに書き込みあるいは読みだしができる。これは画像メモリとRI-SCore間のロードストア命令がピクセル当たり4ns以下で実行できることを意味する。

5 Graphics Hardware Modules

グラフィックハードウェアモジュールを構成するそれぞれのプロセッサはMIMD並列処理構造からなり、図3に示すようにRISC coreとピクセルキャッシュ間に位置する。これらのうち、Rendering Processor, Shading Processor, HSR (Hidden surface Removal) Processor, Texture Mapping (Reverse Projection) Processor はそれぞれ並列あるいはパイプラインプロセスとして、また BitBlit Processor, Gaseous Object Generator, Video Input

/Output Controllerはそれぞれ独立して動作する。これらプロセッサで処理されたすべてのピクセルは画像メモリに書き込まれる前にバスマルチプレクサを介して2次元あるいは3次元配列でピクセルキャッシュあるいはピクセルキャッシュFIFOにロードされる。ハードウェアモジュールはすべて65MHzで動作し、それぞれのプロセッサ内のすべての処理はフィードバック系の無いパイプラインに構造化されている。

5-1 Rendering Processor

Rendering Processorはポリゴンの頂点座標情報から外郭線上のすべての点の情報を求めるものであり、2組の線形補間回路（DDA）が用いられる。一組の線形補間回路内には11個のDDAが含まれる。それらは3次元座標値 x, y, z 、テキストチャマッピング座標値 u, v, w 、法線ベクターより算出された頂点座標の水平垂直傾き角度成分:

N_h, N_v 、光源法線 L_h, L_v およびバンプ法線アップベクター N_u である。よって3次元のポリゴン塗りつぶし処理においては、全体で22個のDDAが同時並行してこれらの成分を各頂点間で補間することになる。角度成分 N_h および N_v はPhong Shadingに用いられ、Gouraud Shadingでは N_h 用DDAが輝度成分に代わり、 N_v は使用されない。ReflectionおよびRefraction法線 R_h, R_v は L_h, L_v とリプレースされる。2組の線形補間回路はそれぞれポリゴンの底辺から左右に分岐して補間を開始する。この補間速度はピクセル当たり15nsで進行し、Y座標が一致する毎に左右の（すなわち水平軸と交差する）端点を次段のShadingプロセッサに出力する。よって左右それぞれのアウトラインの傾きがY軸寄りである場合には、両端点の座標値およびその属性22種すべてのデータは15nsで求められる。この外郭線補間はポリゴン形状に制約されない任意の形状が可能である。

5-2 Texture and Video Mapping Processor

図7にマッピングおよびシェーディング回路構成を示す。テキストチャマッピングはRendering Processorから与えられた x, y 座標値以外に3つのパラメータ u, v, w によって求められる。ここで u および v はそれぞれデバイス座標変換後のテキストチャパターンの定

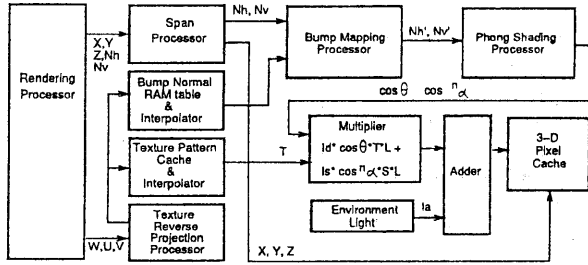


図7 Shading Processing System

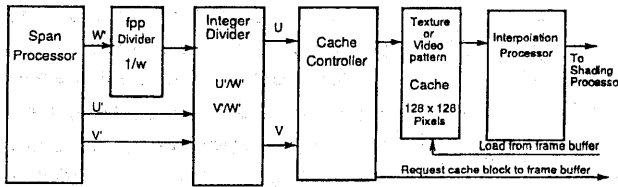


図8 Reverse Projection Processing System

義座標であり、 w は z 成分を含む homogenous 座標値である。Reverse Projection Processor ではこれらの値を水平軸に沿って線形補間(スパンプロセッサ)すると共に、透視変換による非線形変換に対して逆写像変換を行う。これには u, v 座標を w 成分で除算することが必要であり、これを画素当たり 1 DDA クロック以内で処理を行うために w 値のみ浮動小数点変換値による除算を行い、その後 u, v 座標との整数乗算を行う。この関係を図 8 に示す。マッピングするソースパターンは画像メモリに記憶される。その一部がパターンキャッシュにロードされる。パターンキャッシュは逆写像変換された u, v 座標値およびその小数点部によってアクセスされる。この際、キャッシュミスが発生すると画像メモリから 3 2 画素単位で所定の領域のパターンがパターンキャッシュにロードされる。キャッシュヒットの場合はスパンプロセッサの補間に同期してパターンがキャッシュから読みだされ、フィルタリングの後に、所定の面にマップされる。Truga001 ではテキストチャパターンのフィルタおよびスケールは SINC 関数を用いた 3 次 Cubic Convolution[7] を用いており、この場合もハードウェア方式では初めて補間参照画素 16×16 ピクセルから 1 ピクセルを 1 クロック以内で補間する構造を開発した。このフィルタ回路を図 8 に示すが補間点と参照画素との関係は式 1 で与えられる。図 9

$$Q(u_j) = \sum_{i=0}^3 f_i(u) Q_i^j \quad Q_i^j = \sum_{j=0}^3 f_j(u) P_{ij}$$

では 3 次 basis 関数を RAM テーブル化し、表 1 に示す関数を記憶することによって、ソースパターンの画質に応じ線形変換から 3 次スプラインまで各種のフィルタリングを可能としている[8]。

Truga001 では 30 あるいは 60 Hz でリフレッシュされる TV 映像の 3 次元ポリゴン面へのビデオマッピング機能も内蔵する。テキストチャマッピングとビデオマッピングの違いはビデオ入力の取り込みとレンダリングプロセッサとの同期処理が追加されることである。図 1 に示したようにこのチップはビデオ入力系と、レンダリング系とは互いに独立しており、ピクセルキャッシュの後段の FIFO においてのみバスを共有する。この結果レンダリングとビデオ信号の取り込みに関して 73 MHz HDTV においても 15% 程度の性能低下(後節参照)しか生じない。ビデオ映像マッピングの回路構成を図 10 に示す。

関数	$f(u)$	補間
1st Order	$u + 1$	$-1 \leq u < 0$ Triangle
	$-u + 1$	$0 \leq u < 1$
	0	$1 \leq u$
2nd Order	$-u^2 + 3/4$	$0 < u < 1/2$ Bell
	$(4u^2 - 12u + 9)/8$	$1/2 \leq u < 3/2$
	0	$3/2 \leq u$
3rd Order	$(3u^3 - 6u^2 + 4)/6$	$0 \leq u < 1$ B Spline
	$-(u - 2)^3/6$	$1 \leq u < 2$
	0	$2 \leq u$
Sinc	$1 - 2u^2 + u^3$	$0 \leq u < 1$
	$4 - 8u + 5u^2 - u^3$	$1 \leq u < 2$
	0	$2 \leq u$

5-3 Bump-mapped Phong Shading Processor

Phong Shading Model は面と光源の法線を用いて、面上における反射角および反射光が視点に入力する方向余弦からそれぞれ拡散および鏡面反射成分を求めるものである。

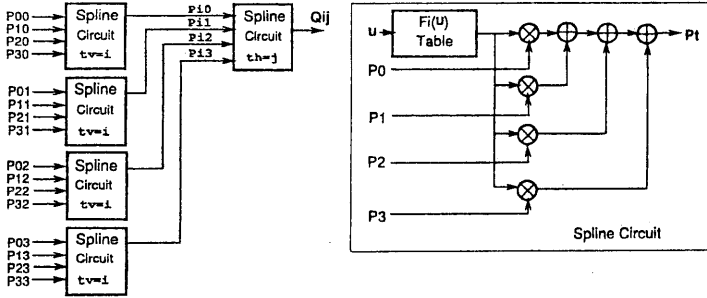


図 9 Spline Interpolation Circuit

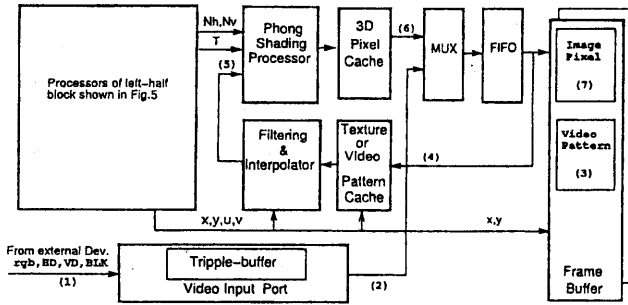


図 10 Video Mapping Processing System

この関係を図 1 1 に示す。従来この反射計算には法線ベクトルが使用されており、この結果補間毎(ピクセル)に正規化処理を必要とし、そのための平方根や除算処理等の計算コストが実時間処理を阻む要因となっていた。Truga システムでは法線を視点座標に対して水平および垂直角の極座標で表現する。これは式 2 に示すように反射光および反射光と視点との方向余弦が、水平あるいは垂直角のいずれか一方を変数とする三角関数項で表現でき、ベクトルではなく角度を扱うため補間後の正規化処理を必要としないことに着目したからである。

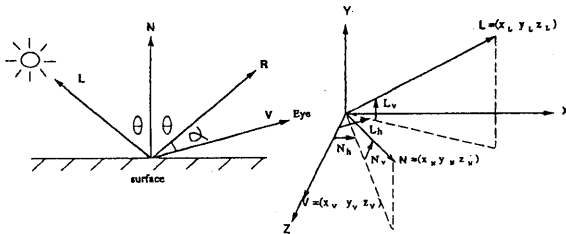


図 1 1 Phong Shading Model

$$I = I_a + I_p = I_a + I_d \cos \theta + I_r \cos^2 \alpha$$

$$\cos \theta = \cos N_v \cos L_v [\cos(L_h - N_h) - 1] + \cos(L_v - N_v)$$

$$\cos \alpha = 2 \cos N_h \cos N_v \cos \theta - \cos L_h \cos L_v$$

ここで I_a , I_d , I_r , N_h , N_v , L_h , L_v , n , θ , および α はそれぞれ環境光、拡散係数、鏡面反射係数、面水平角、面垂直角、光源水平角、光源垂直角、鏡面反射指数、面と光源法線との反射角および反射光と視点軸との角度である。従来 Angular Interpolation による Phong shading 方式の提案 [9] があるが、光源と面法線との差分値をポリゴンの両端点で線形補間するため、バンプマップや光源が面に近接すると誤差を生じる問題をもっていた。本方式はそれぞれの法線成分が関数内で独立した関係にあり、そのような問題を含まない。以上から、Phong Shading 回路は式 (2) より図 1 2 となる。

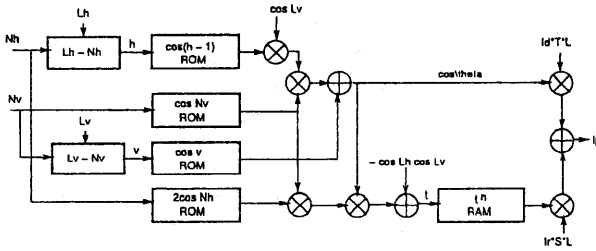


図 1 2 Phong Shading Circuit

この回路では平行光源ではフレーム当り L_h, L_v はコンスタントであり、またスポットライティングでは面法線同様に補間された L_h, L_v が回路に与えられる。一方、バンプマッピング[10]は表面の粗さを、光源の明暗によって表現するもので、面の法線をそれぞれの点で定めたバンプ法線によって回転するものである。このシステムではバンプマップパターンをテクスチャーマッピングと同一の座標系に定義する。よってバンプ法線はテクスチャパターン同様にRAMに記憶し、これをテクスチャアドレスで読みだす。テクスチャ同様にフィルタリングは不可欠である。このバンプマップ法線もテクスチャ u, v 座標と垂直な軸に対する水平および垂直角 B_h, B_v で定義する。バンプマップと面法線との関係は図1.3となる。回転後の法線 $N_h' N_v'$ は下式で与えられる。

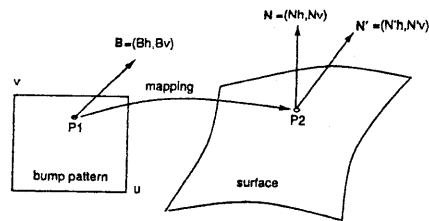


図1.3 Bump Mapping Model

このベクトルを N_r とすると B_x, B_y, B_z との関係は式4.4となる。

$$\begin{aligned} N_v' &= \arcsin[\sin|N_v + A_z| \sin|A_x|](-1)^{s_1} \\ N_h' &= \arcsin(\cos A_x \sqrt{1 - \sin^2 N_v'})(-1)^{s_2 + s_3 - 1} - s_3(-1)^{s_3} \pi + N_h' \\ A_x &= \arccos(\sin|B_h| \cos B_v) s_0 \\ A_z &= \arcsin(\sin B_v / \sin|A_x|) \end{aligned}$$

$$\begin{aligned} s_0 &= \begin{cases} 1 & \text{if } B_h \geq 0 \\ 0 & \text{otherwise} \end{cases} & s_1 &= \begin{cases} 1 & \text{if } N_v + A_z \geq 0 \\ 0 & \text{otherwise} \end{cases} \\ s_2 &= \begin{cases} 1 & \text{if } |N_v + A_z| \geq \pi/2 \\ 0 & \text{otherwise} \end{cases} & s_3 &= \begin{cases} 1 & \text{if } A_x \geq 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

以上からバンプマッピング回路は図1.4となる。ここでバンプ法線成分 B_h, B_v はバンプパターンとしてメモリに記憶されるが計算簡単化のため A_x, A_z を記憶する方が有利となる。図1.3および図1.2に示すバンプマッピング回路と Shading 回路をそれぞれパイプライン接続する構成とは異なり、回転マトリックスを用いた方式も考えられる。Truga001はこの方式を採用しており、そのアルゴリズムはバンプ法線と面法線との回転法線をそれぞれの x, y, z 軸成分で N_x', N_y', N_z' とし、

$$\begin{aligned} \begin{bmatrix} N_x' \\ N_y' \\ N_z' \end{bmatrix} &= \begin{bmatrix} \cos N_h & 0 & -\sin N_h \\ 0 & 1 & 0 \\ \sin N_h & 0 & \cos N_h \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos N_v & \sin N_v \\ 0 & -\sin N_v & \cos N_v \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} \\ \cos \theta &= L_x N_x' + L_y N_y' + L_z N_z' \\ \cos \alpha &= 2 N_z' \cos \theta - L_z \end{aligned}$$

面と光源との反射角 θ および視点と反射光との角度 α とは下記となる。一方、バンプ法線をテクスチャ座標で定義する場合、面の法線を中心とする回転に対して光源に対する回転の影響を受けない問題が生じる。このためバンプ法線による明暗が形状の表現をもつ場合には、Up-vector の設定が必要である。

$$\begin{bmatrix} B_x' \\ B_y' \\ B_z' \end{bmatrix} = \begin{bmatrix} \cos N_r & -\sin N_r & 0 \\ \sin N_r & \cos N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}$$

ここで

$$\begin{aligned} B_x &= \sin B_h \cos B_v, & B_y &= \sin B_v, & B_z &= \cos B_h \cos B_v \\ L_x &= \sin L_h \cos L_v, & L_y &= \sin L_v, & L_z &= \cos L_h \cos L_v \end{aligned}$$

以上から Bump-mapped Phong shading 回路は図1.5で与えられる。図1.5の回路に使用するメモリは鏡面反射指数テーブル⁶⁾を除いてすべてROMで構成できる。鏡面反射指数テーブルは16ビット入力を8ビットで2分割した線形補間法を、またROMは1Kワードで出力ビットを12ビットとしている。このため1万ゲート以内で図1.5のすべての回路が実現できる。多光源で、スポットライティングの場合には L_h, L_v は面法線 N_h, N_v 同様にそれぞれのポリゴン頂点に定義され、これを補間するため、図1.6に示すDDA回路を用いる。この回路はクロック毎に光源毎の L_h, L_v を生成し、図1.5の L_h, L_v に出力する。拡散および鏡面反射光はそれぞれ、拡散係数、テクスチャカラー、光源色と、また鏡面反射係数、鏡面色、光源色と乗算され最終の輝度を得る。Truga001では図1.5に示す回路を4ライン分内蔵する。この結果、それぞれが1.5nsの動作速度で並列処理されるため、ピクセル当たり、3.8nsで shading されたピクセルがピクセルキャッシュに記憶される。ベクター描画においてはこの4回路のうち一つだけが使用される。

5-4 Shadow Casting

光源の投射によって生じる形状自身もたらず陰影

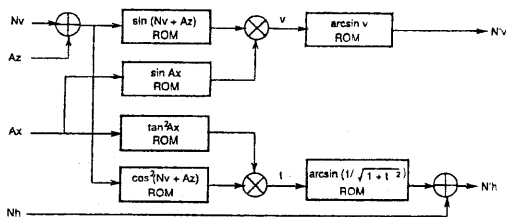


図 1 4 Bump Mapping Circuit I

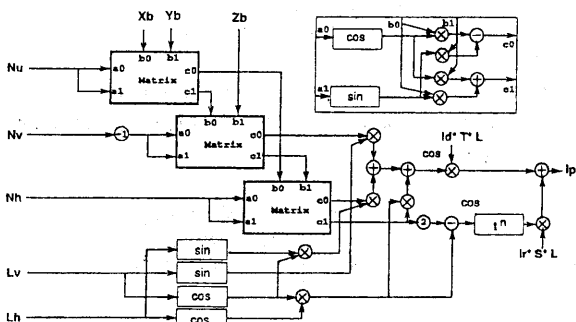


図 1 5 Bump Mapping Circuit II

は、Shading 処理によって生成されるが、他の物体に及ぼす陰影の投影処理は特定のプロセスが必要となる。これらのハードウェア化については幾らかの方式が知られているが[11]、Truga001では光源座標軸に対する投影図形の決定と、視点座標軸における投影図形とのマッピングから成る2パスZバッファ方式を、物理的に同一の画像メモリを共用する方式とともに導入している。この方式の問題点は視点軸と光源の投影軸とが一致しない場合、陰影の描画有効範囲が画像メモリの容量に影響されることである。この結果Z方向に長く横たわる図形の陰影を視点軸と90°の方向から光源で照射した場合、陰影の範囲を満足させることができない。最悪の場合は光源軸が視点座標軸に直交する場合、陰影の視点方向の有効範囲は1K-2Kピクセル程度となる。これを避けるため、このシステムでは陰影形状を投影する画像メモリに対してはZ成分に反比例する動的なスケールングを書き込み形状に対して行い、視点から離れる程曖昧さを増す一方、有効範囲を広める方法をとる。使用する物理的画像メモリサイズは2K x 2Kである。

5-5 Gaseous Object Renderer

Gaseous Object Renderer はガス状イメージの生成回路である。霧、煙、雲等の表示にはテキストチャタートンを用いる方法と、ビデオカメライメージによる合

成がある。また霧の表現にはカラーラックアップテーブル方式がある。これらに用いられるオブジェクトはすべて2次元図形であり、ウォークスルー仮想現実空間でしばしば生じる視点が物体内に移動する描画には問題となる。このためTruga001ではガス状物体をボリュームレンダリングによって、3次元空間内に生成し、これらに透明感、Turbulentモデルに基づく動的変化を与える。これらガス状物体の生成には通常下記の処理および条件が必要である。

* Turbulentモデルに基づく不規則な形状変化

* 透明感およびハイライティング

* 面定義物体との3次元空間内での合成
このプロセッサではオブジェクトは階層的なセグメントで構成し、その原始セグメントの位置と形状はガウス乱数によって生成される複数の粒子で定義する。全体イメージはそれぞれのセグメントが集合したローカルオブジェクトと、更にローカルオブジェクト(小単位のガス状物体)を集合したグローバルオブジェクトで構成する。この関係を図16に示す。それぞれのセグメントはそれぞれ一つのあるいは連続する参照点を持つ。この参照点は3次パラメトリック関数のノンユニフォームな線上に定義されており、この曲線の制御点の移動によって参照点も移動する。同様にグローバルオブジェクトを構成するローカルオブジェクトもまた、それぞれパラメトリック関数曲線上に位置付けられる。ガス状物体の全体定義のためのハードウェアは下記で構成される。

- * ガウス乱数発生器
- * パラメトリック自由曲線発生器
- * 原始画素発生器
- * マトリックス乗算器
- * 法線ベクタ定義回路
- * Phong Shading回路
- * 陰面消去回路
- * 透明感生成回路
- * 平均化フィルタ
- * 面定義物体合成回路

上記のハードウェアユニットの内、後者3回路はTruga002に内蔵される。これらの全体構成を図17に示す。ガス状物体は2つのモデラーによって定義する。それらはオブジェクトモデラーとモーションモデラー

である。オブジェクトモデラーは物体の形状を定義するもので上記の乱数発生による粒子の分布および密度また、その基本形状と属性を決定する。

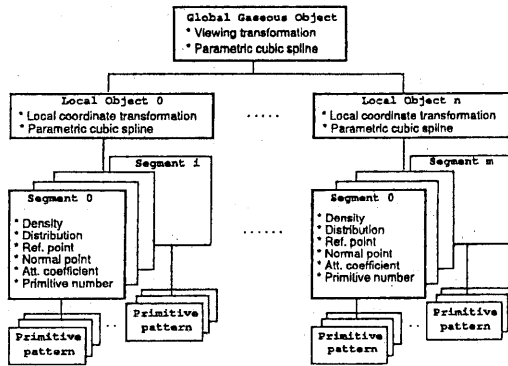


図16 Gaseous Object 構造

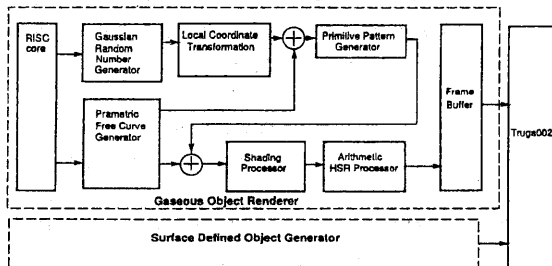


図17 Gaseous Object Renderer 全体回路構成図

一方、モーションモデラーは定義された物体の動きを定義する。形状はいずれもモデラーに密接に関係する。図17のプロセスでは、ガウス乱数発生器において3次元座標値を生成し、この座標点に対してマトリックス乗算器において視野変換を行う。変換後の各座標点には所定のプリミティブ形状が付加される。これらプリミティブにはハイライト効果を出すためにセグメント毎に与えられた法線計算のための参照点が定義され、プリミティブ毎に法線を求める。この法線は Shading で述べたように、2つの傾き成分として定義する。それぞれのプリミティブがそれぞれの傾き成分をもつことによって、Shading回路において拡散反射成分が求められる。画像メモリにガス状画素を記憶する際、通常の面定義物体画素と異なる点は、陰面消去回路において同一の視点軸に対して発生したピクセルの書き込み回数(重なり数)を計数し、輝度と共にこれをピクセル毎に画像メモリに記録し、透明感表示のデータとして用いることである。画像メモリに記憶されたこれらデータは、表示モニターの走査順に従って

読み出され、Truga002に与えられる。Truga002では輝度、前記の重なり回数、セグメント毎に設定された輝度減衰率等のデータを平均フィルタリングをした後、面定義物体との合成を行う。よってこのシステムでは面定義物体とガス状物体とはそれぞれ異なるTruga001モジュールで計算し、これらをTruga002で合成する方法を基本としている。一組のTruga001で両処理を行うことも可能であるが、この場合、ガス状物体のフィルタリングも合成以前にTruga001で行う必要がある。

5-6 Video Image Controller

Truga001は表示モニターへのビデオリフレッシュ用ポート以外に双方向ビデオカメライメージ入出力ポートを持つ。これらポートよりHDTVあるいはNTSCのR,G,B形式の外部同期方法によるデータの取り込みあるいは出力を可能にする。入力は最大受信周波数110MHz、また出力は50MHzである。取り込みデータはFIFOおよびピクセル間演算回路を通して画像メモリに記憶されるがメモリの記憶場所はアプリケーション側からの設定による。画像メモリ

アドレスの制御は外部水平同期信号で行う。ビデオデータの画像メモリアクセス占有時間比率はHDTVにおいても全体の15%程度であり、レンダリングプロセスと同時に並行してもフレーム落ちの無いビデオCapturingおよび同時表示が可能となる。またRISCcoreからビデオ画素に関して任意のZ値定義が出来るため、優先順位を持つビデオ映像間あるいはCG間でのスーパーインポーズ効果を得ることが出来る。

5-7 Pixel Cache and ALU[12]

Truga001ではレンダリングあるいはビデオカメラ画素はピクセル間演算を実行する前にピクセルキャッシュあるいはFIFOに記憶される。ここでは単にデータをバッファリングするだけでなく、画像メモリに書き込むための2次元(ピクセルビット数を考慮すると3次元)あるいは1次元配列変換を行う。この多次元配列構造はデータによって異なり、ベクタ、ポリゴンでは8(X) x 4(Y) x 24(Z) x 24(Pixel)の3次元配列を、BitBlitあるいはビデオカメラデータでは32(X) x 1(Y) x 24(Z) x 24(Pixel)の2次元となる。

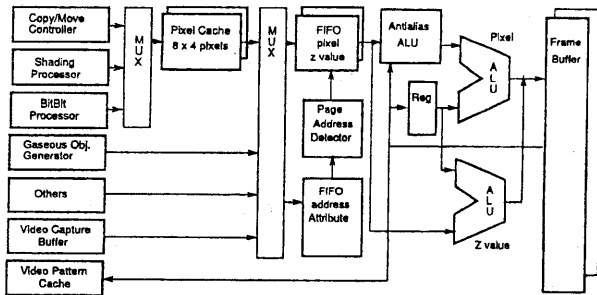


図 1 8 Pixel Cache and FIFO構造

このような配列をもつ32ピクセルをDAMの場合Z値を含め、約180nsで画像メモリに書き込むために、画像メモリは特別な構造をもつ。すなわち、一つのプレーンは4つのサブモジュールからなり、一つのサブモジュールは8ビットのバスで構成されなければならない。8x4ピクセルキャッシュに対して8ビットバスは水平8ピクセルを、4つのモジュールは垂直4ラインを受け持つことを意味する。この結果、一サイクルで32ピクセルの並列アクセスが可能となる。ベクタあるいはポリゴン描画ではShading Processorにおいて生成されたXY座標値の下位ビットがピクセルキャッシュのアドレスとなる。よってベクタ描画ではその傾きによってキャッシュヒット率が変化する。ポリゴンの塗りつぶし描画では前節で述べたように、4組の補間回路が4ラインを15nsで同時に実行する。水平軸キャッシュサイズは8ピクセルであるため、約120nsでキャッシュフルとなる。このシステムでは画像メモリへの陰面消去用Zバッファ書き込みを含めたキャッシュ全体の書き込みサイクルタイムの平均値はこの値にほぼ等しいため、画像メモリには等価的に4ns/ピクセルとなる。ピクセルキャッシュは2つのバッファ構造からなる。それらはキャッシュレジスタとFIFOである。FIFOは3組あり、それぞれ16ワードの深さを持ち、一ワード当たり32x24ビットのピクセルおよびZ値、32ビットのアドレス用である。キャッシュレジスタはダブルバッファリング構造を持つ。この構造はキャッシュミスが発生した場合のキャッシュからFIFOへの転送損失をゼロにするためである。FIFOの深さを16ワード以上にしても転送の改善はほとんど(10%以内)見られない。図18にキャッシュおよびその周辺回路を示す。図に示すように、ShadingプロセッサおよびBitBlitからのデータはピクセルキャッシュを通してFIFOに与えられるが、ビデオカメラデータからのデータは直接マルチプレクサを通してFIFOにロードされる。

これは後者がそれぞれのモジュール内に独自のキャッシュ(バッファ)を持っているためである。この場合後者からのデータは32ピクセル毎にスライスされてFIFOにロードされる。より高速な画像メモリへのアクセスを計るため、画像メモリへの書き込みの際にキャッシュブロックのアドレスをFIFOから先読する。これら連続するアドレスにおいて画像メモリのカラーアドレス

相当に変化のない場合そのキャッシュブロックデータは同一ページ内に位置付けられるため、ページアクセスを画像メモリに対して実行する。この結果ページモードの有無で約25%のアクセスタイムの改善が得られる。FIFOからのデータは画像メモリへの書き込み時、デスティネーションピクセルとの論理的あるいは算術的演算が行われる。この際、Truga001ではDistance法によるAntialiasを行うことができる。これらの演算は32ピクセルが同時並行して実行される。

5-8 Frame Buffer

グラフィックシステムの性能を決定する要因の一つとして画像メモリのバス幅が取り上げられる。特種RAMが開発製品化されているものの、ASIC技術の進歩と比較するといまだ、この要因は大きな問題点となっている。Truga001ではその設計概念としてビデオRAMではなく、通常のシングルポートDRAMを画像メモリの標準としている。これは画像メモリには表示画像情報だけでなく、リアルタイムビデオカメラ情報、テキストチャッピングパターン等の記憶、更に画像処理のためのピクセルワーキング領域、Zバッファ等に加えて、動的表示のためのこれら情報のダブルバッファリングを考慮するとモニター表示領域の少なくとも4倍以上の領域が必要であり、コストパフォーマンス上の問題がその背景である。このシングルポートDRAMの問題点はバス幅が狭いことである。これを解決する方法は多数のDRAMを使用して物理的なバス数を増やす方法と[13]、Truga001が導入している多次元配列構造の一括アクセスのいずれかとなる。自然色表示での画像メモリは通常24プレーン構成となる。Truga001チップは384本のラインを持ち、2クロックで24プレーン(32ピクセル分)を直接アクセス出来る。画像メモリは標準で4Kx4Kx24bitで構成する。16MbitDRAMを使用する場合、自然色では24個のDRAMとなる。この構成であれば1280x1024あるいはHDTVの1920x1035モニターに対応す

るピクセルとZバッファのダブルバッファリング構造が可能となる。シングルポートDRAMにおいてもっともコストパフォーマンスに優れた素子は同期式DRAMと考えられる。この素子の場合、ピクセルの書き込み読みだしにはランダムカラムアクセス（2バースト、3カラムアドレスレイテンシ）を、ビデオリフレッシュには2つの8バーストレングスモードを用いる。

5-9 Video Refresh and Cursor Generator

Truga001には256x8ビットのビデオリフレッシュ専用のシフトレジスタが内蔵されており、画像メモリの走査読みだし情報をシリアルビデオ信号に変換する。この信号はカーソル発生器を通してD/Aコンバータに出力される。出力は2相の信号となる。カーソル発生器は2つのビデオカーソル：クロスヘアー、レクタングルと、4つの書き込み可能なアプリケーション定義形状カーソルがサポートされている。これら再定義可能なカーソルは同一スクリーンにおいて同時表示が可能である。

6 性能評価

Truga001は標準構成では同期式DRAMを使用する。シングルポートDRAMを用いる場合はメモリのシステムサイクルを定義する。このサイクルはモニターのビデオ周波数に依存する。110MHzにおいては2.33μsとなる。一方、画像メモリとのアクセスはデータの書き込み読みだしおよびビデオリフレッシュがあり、Z値を含む画像メモリへの32ピクセルの書き込みサイクルは180nsとなる。この中にはローおよびカラムアドレスリング、2バーストピクセルおよびZ値Read-ModifyWrite転送、Z値比較および転送（60ns）を含む。一方、ビデオリフレッシュサイクルは220nsとなる。よって、一つのシステムサイクルの中には一つのビデオリフレッシュサイクルと複数のピクセル書き込みあるいは読みだしサイクルが含まれる。80MHz以上の同期式DRAM（例えば100MHzDRAMを75MHzで使用）を使用した場合での、性能評価結果を図19 a-cに示す。それぞれベクタおよびポリゴンでの特性であり、ベクタの傾きおよび始点はキャッシュに対してランダムに、またポリゴンは正三角形とし、その原点もランダムに与えるものとする。それぞれパターンサイズがキャッシュサイズを超えたテクスチャマッピングおよびビデオマッピ

ング時の特性である。

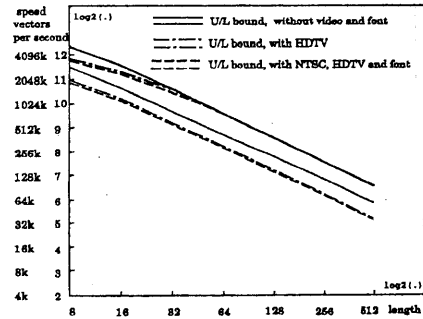


図19 a ベクタ描画特性

ベクタおよびポリゴンそれぞれの描画時に対してHDTVのビデオ取り込み、NTSC出力および文字描画（16x16ドット2000文字/秒）が同時並行して処理される場合の性能も示している。図19 aより、ベクタ性能は最大4百万（10ピクセル）/秒である。U/Lは最大/最小性能を示す。これはベクタの傾きおよび始点位置に依存する。ポリゴンは1.2百万（100ピクセル）/秒となる。ポリゴン性能はいずれも隠面消去、テクスチャマッピング（パターンサイズキャッシュ内）、バンプマッピング、フォンシェーディングを併合した実行時である。特性から、HDTV（73MHz）取り込みと表示を同時に実行した場合、レンダリング性能は10-15%の低下しか見られない。秒間2000文字程度の表示はほとんど性能に影響を与えない。これはそれぞれが独立した回路および入出力ポートで処理されているためである。

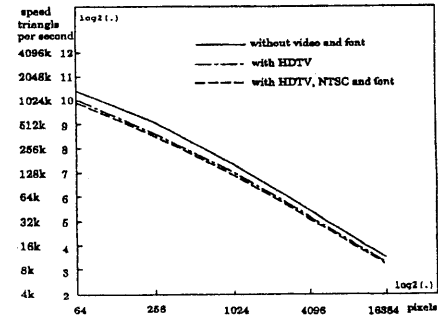


図19 b ポリゴン 描画特性

唯一FIFOにおいてこれらは競合（フル状態における待ち）するが、これは画像メモリバンド幅に起因するものである。

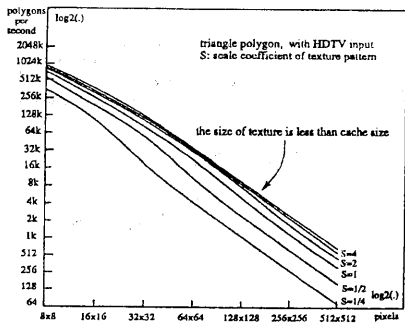


図19c Video mapping 描画特性

表2にはそれぞれのモジュールあるいはプロセッサの回路規模を示す。また付記には特性を記す。表におけるGRID数は2層配線時であり、ゲート数では全体で94万ゲート相当となる。

表2

Processor	Grid(1,000 units)
RISC core with FPP	150.5
Rendering Processor	35.7
Shading Processor	180.2
Pixel Cache	780.0
Texture Mapping Processor	470.6
Gaseous Object Generator	30.9
Image Processor	71.8
BitBlit Processor	25.7
Video I/O Controller	55.9
Cursor & Video Timing Generator	78.3

7 むすび

この研究は各種のグラフィックス描画機能を包括するポリゴンレンダリング用LSIとして、GVIP継続プロジェクトの一貫として行われたものである。GVIPプロジェクトから起算して6年の開発期間を要している。本研究が示すように特定のグラフィックスプロセッシング領域をターゲットとするのではなく、インテグレートされた全体構造からアプローチする研究例は大学では非常に少なくその多くは特定の構造研究を取り上げるのが一般的である。しかしながら我々は、新しいアルゴリズムの開発によって、一部の機能が極めて優れた性能をもつことができても全体性能では期待する

ほどの効果が得られない実用上の問題例を数多く経験する。この結果 Truga001に見られるように1つのLSIを設計するのに必要十分な機能とともに、それらの相互関係も最適化し、全体にバランスのあるプロセッサシステムを開発することが一層重要である。本研究ではLSI化としては4つの初めてと思われる回路開発が成された。Phong shading回路、bump mapping回路、1クロック演算cubic splineビデオおよびテクスチャマッピング回路、およびgaseous object rendererである。これら回路の実験データの詳細は別の論文で取り上げることにする。

文献

- 1) Potmesil, M., Hoffert, E.M.: The Pixel Machine: A parallel Image Computer. ACM SIGGRAPH'89, 23, 3 pp.69-78(1989)
- 2) Silicon Graphics Inc.: Power Series. Technical Report(1990)
- 3) Fuchs, H., Poulton, J., et al.: Pixel-Plane5; Heterogeneous Multi-processor Graphics System Using Processor-Enhanced Memory. ACM SIGGRAPH'89, 23, 3, pp.79-88(1989)
- 4) Jayasinghe JAKS, et al.: Two level pipeline of systolic array graphics engines. In: Advances in computer graphics and CAD. Springer, Berlin Heidelberg, pp.133-148(1991)
- 5) Gharachor, N., et al.: Subnanosecond Pixel Rendering with Million Transistor Chips, ACMSIGGRAPH '88, pp.41-49(1988)
- 6) Ikedo, T., A Scalable high-performance graphics processor: GVIP. The Visual Computer, 11, pp.121-131(1995)
- 7) Bernstein, R.: Digital Image Processing of Earth Observation Sensor Data. IBM J. RES Deveop., pp.40-57(Jan. 1976)
- 8) Patt, W.K., Digital Image Processing(2nd edition). Chap4: Image sampling and Reconstruction. John Wiley & Son Inc.
- 9) Kuijk A.A.M., and Blake E.H., Faster Phong Shading via Angular Interpolation. North-Holland, Computer Graphics Forum 8, pp.315-324(1989)
- 10) Blinn, J.F., Simulation of Wrinkled Surface. ACMSIGGRAPH '78, 12, 3, pp.286-293 (Aug. 1978)
- 11) Forley J., et al: Computer Graphics: Principles and Practice (2nd Edition), Addison Wesley (1990)
- 12) Ikedo T., Design and Performance Evaluation of a Pixel Cache Implementation with application specific circuits. The Visual Computer, 12, 5, pp.215-233(1996)
- 13) Akeley K., Reality Engine Graphics. ACMSIGGRAPH'93, pp.109-116(1993)

付記

機能特性表

項目	特性	注
Scalability	Theoretical unlimited	
Embedded RISC core	32-bit, 64-register file 8K/8K instruction/data cache Pixel register files	75MHz
Floating Point Processor	四則演算、平方根 64-register file	150Mflops
Shading		Hw
Gouraud Shading	Intensity DDA	
Phong Shading	Angular Interpolation	
Spot lighting	Angular Interpolation	10 sources max.
Bump Mapping	Angular Interpolation	Hw
Embedded Cache	64 x 64 x 20-bit	
Pattern size	2K(u) x 2K(v) max.	Frame buffer
Interpolation	Linear, B-spline, sinc, etc.	Selective
Texture Mapping		Hw
Embedded Cache	128 x 128 x 24-bit	
Pattern size	2K(u) x 2K(v) max.	Frame buffer
Interpolation	Linear, B-spline, sinc, etc.	Selective
Video Image Mapping	Real time mapping	Hw
Pattern size	2K(u) x 2K(v) max.	Frame buffer
Interpolation	Linear, B-spline, sinc, etc.	Selective
Reflection Mapping	Angular Interpolation	Hw
Refraction Mapping	Angular Interpolation	Hw
Anti-aliasing	Distance method/Sinc filtering	Hw
Z buffering	24-bit(19-bit fract./5-bit Exp.)	Hw
Shadow Casting	2-pass z-buffer method	2K x 2K buffer
Alpha Blending	Linear/non-linear	Hw/Truga002
Gaseous Object Renderer	Gas, Fog, Cloud, Steam, etc.	Microprogram
Dithering		Microprogram
Font-fill		Hw/RISC core
Video Capturing	HDTV, NTSC, etc.	
Video Output	2 ports	150/50MHz
Picture in Picture	30Hz(1000 x 1000)	Hw
BitBlt	1,2,3 D pixel format	Hw
Cursor	6-video cursor	

Hw: ハードウェア回路構成