

## 解説



## 三次元グラフィックスの動向と技術課題†

福永 泰†† 藤田 良†† 古賀 和義††

## 1. はじめに

昭和60年に開催されたつくば万国博覧会で、計算機が生成した三次元映像はマスコミに大きく取り上げられ、展示館の前に長い行列ができた。その後、各種のテーマパークでも、特殊なメガネをかけて、立体視を体験できるコーナーが設けられ、観客の喝采を浴びている例も多い。一方では、映画や、広告の世界でも、三次元グラフィックスを駆使し、映像との一体化技術により、アピール性を向上させている。このように、三次元グラフィックスを用いた映像技術は、特に、エンタテインメントの世界を中心に、一般の人の目に触れる機会が多くなってきている。

ところで、こうした映像は、あらかじめ、多大な時間と、膨大な計算機を使用して作られたものをフィルムやビデオテープなどに記録したもので、その生成時間は、膨大なものになっており、また、再生時に、ストーリーを変更することはできない。

一方、専門的な分野では、主に、機械系のCADシステムや、シミュレーションを中心とする数値演算結果の可視化などの分野で、高速な三次元表示機構を用いたシステムが1970年代ぐらいから盛んに使用されていた。この世界では、数千万円から数億円におよぶコストの高いシステムを、ときには一人で占有し、会話的に画像を生成し、設計作業を効率化している。このため、利用できる人は、マーケットの大きい分野の開発業者などに限られていた。

ところが、最近のマイクロプロセッサの高速化技術はめざましいものがあり、ここ7、8年間の性能向上の割合は、年率1.8倍にもおよぶ。この

まま推移すると、たとえば、西暦2000年以前に、数十万円で、1GIPSの性能を有するパソコンやワークステーションを手にもすることも予測できるようになってきた。これは、現状1000万円を越えているシステムの性能よりも、さらに一桁高速なシステムを、一般家庭や、ビジネスマンが所有する時代が到来することを意味する<sup>1)</sup>。

このような、高速な性能をグラフィックスの分野に生かすと、今までのように、作られた映像を再生するだけで満足していた時代から、リアルタイムでストーリーを変更できる三次元グラフィックス映像を、一般の人でも楽しめる時代が、あと数年で実現できるようになると予想される。実際、最近のゲーム機では、三次元映像を取り扱う例もみられるようになってきた。こうした時代背景をとらえ、ここでは、三次元グラフィックス技術の現状を解説する。

## 2. 三次元グラフィックスの進展

1960年代のユタ大学を中心としたディスプレイ応用技術の進展に端を発し、宇宙や、軍事、CAD応用をベースとしたグラフィックス技術の開発が進められてきた<sup>2),3)</sup>。初期のディスプレイ装置は、直接、ディスプレイ面が表示情報を蓄えるストレージタイプや、表示するベクトルの方向に電子ビームを振らせるベクトルスキャンタイプのディスプレイが中心で、表現できる画像も限られたものであった。ところが、ダイナミックメモリが安価に提供されるようになると、表示する映像を、いったん、画像蓄積用のメモリ(フレームメモリ)に作成し、CRTには常にメモリから読みだして表示するラスタスキャンタイプのディスプレイが、安価に提供できるようになった。

ラスタスキャンタイプのディスプレイは、ストレージタイプよりも、動的な表示に適し、また、ベクトルスキャンタイプより、塗りつぶし表示に

† The Trend and Technical Viewpoint of 3D Graphics by Yasushi FUKUNAGA, Ryo FUJITA and Kazuyoshi KOGA (Hitachi, Ltd. Hitachi Research Laboratories).

†† 日立製作所日立研究所

適することから、三次元動画表示が実現できる時代に突入した。

1970年代のグラフィックスシステムは、ホスト計算機で作られたグラフィックスデータを単に表示だけの端末装置を中心として発展した。大型計算機と端末の間は、専用のインタフェースでも設けなかり、kバイト/秒オーダーのデータ転送しかできず、大型計算機で高速にデータ計算しても、その情報をリアルタイムに端末に伝送することが難しい時代であった。ところが、1980年代に入って、高性能な32ビットマイクロプロセッサが出現し、大規模なシミュレーションソフトも端末サイドで動作できる環境が整うに従い、表示専用の機能をもつ端末と大型計算機で分担していた作業形態から、すべての処理をマイクロプロセッサで実行するワークステーションが出現した。

ワークステーションシステムへの移行により、大型計算機と端末の間を往き来していた大量の図形データを、バスを用いて転送できるようになり、スループットは、kバイト/秒オーダーから、Mバイト/秒オーダーにまで高速化した。さらに、一回の転送も細かく制御できるようになったため、単に、でき上がった図形データを送るだけでなく、ダイナミックに表示データを変更できるリアルタイム性に優れたグラフィックスシステムが提供できるようになってきた。

こうした動きを振り返ってみると、グラフィックス技術は、単に、描画処理の高速化にとどまらず、図形データファイルから、応用ソフトウェアを介して、いかに高速に、ディスプレイ上に画像を表現するかという組合せ最適化技術であるとい

える。すなわち、ディスプレイや、メモリ、マイクロプロセッサなどの基盤技術の進展、回線やチャンネル、バスなどの伝送技術の進展、あるいは、オペレーティングシステムの特長など、多くの技術の進展速度に合わせて、最適な組合せを行うことにより、それぞれの時代のグラフィックス技術が、成り立ってきているように思える。そこで、以下では、各部の必要とする性能とその実現手段を中心に、現状のグラフィックスシステムの概要について説明する。

### 3. グラフィックス処理の概要

#### 3.1 グラフィックスパイプラインの流れ

一般のグラフィックス処理の流れを図-1を用いて説明する。グラフィックス表示を行う応用ソフトウェアは、各応用プログラムごとに有する図形データベース内のデータを解釈し、グラフィックスライブラリのインタフェースにあわせたデータに変換してサブルーチン呼出しを行う。

グラフィックスライブラリ以下の部分がシステムで提供される処理部（グラフィックス部）に当たり、これは、数値演算を中心とする座標変換／輝度計算部と、ディスプレイ上の画素に展開する画素発生（レンダリング）部、イメージデータを保持する画像メモリ、その内容を表示する表示制御部に分かれている。

座標変換／輝度計算部は、あらかじめ与えられている変換マトリクス値に従い、図形の頂点座標の座標変換を行い、さらに、図形が表示エリア内に入っているか、入っていないかのチェック（クリッピング処理）や、与えられた頂点の輝度

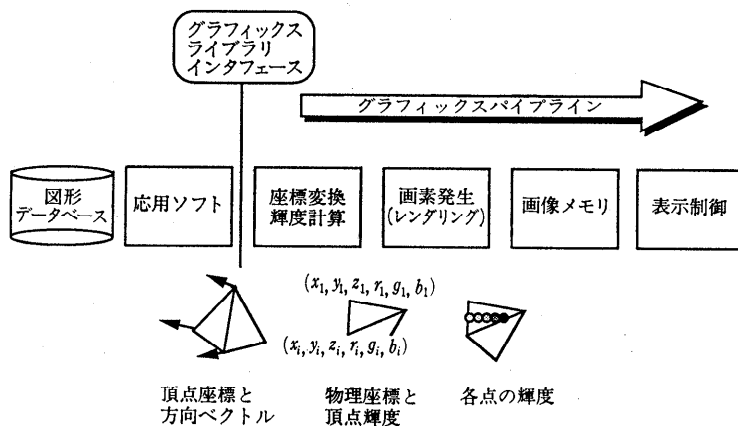


図-1 グラフィックスパイプラインの概要

計算を行う部分である。レンダリング部は、与えられた頂点座標や頂点の輝度値から、塗りつぶし面の内側の画素の値を内挿して求める。こうした処理は、高速化のため、別々のハードで順次行われることが多く、このため、これらの流れをグラフィックスパイプラインと呼ぶ。

演算量などを把握してもらうため、三次元の塗りつぶしコマンドの代表的な処理である連続三角形に焦点を当てて、もう少し詳しく説明する。

図-1の下に示すように、応用ソフトウェアからは、連続三角形の図形データの頂点データが、各頂点の方向ベクトルなどで与えられる。こうした情報量は、一つの三角形当たり、100バイトのデータにおよぶ場合もあり、たとえば、1Mポリゴンの絵で構成される一画面を送ろうとすると応用ソフトからグラフィックス部に100Mバイトものデータを送出する必要がある。

座標変換部では、一頂点当たり、数百命令以上の数値演算処理により、物理座標系への変換や、各頂点の輝度情報が算出される(命令数は、そのときの属性情報によって変わるが、ここでは簡単のため、上記のように仮定した)。

図形が、描画エリアのなかに入っている場合は、レンダリング部で、画素単位の輝度値の算出や、Z値の比較が行われる。この処理内容を、図-2に示す。三角形の各頂点の値(X, Y, Z, R, G, B)はすでに求められているため、まず、左右の辺の座標値、輝度値を順次求める。たとえば、点Pの値は、AB間のベクトルの内挿計算をすることで図-2に示すように求めることができる。P, Qの値を求めると、つぎに、同じラスタ上の各画素の値も同様に、一画素分の変化分を算出し、順次Pに加算していくことで求めることができる。

この操作を、各ラスタに対して繰り返し行うことで、三角形の塗りつぶし図形を描画することができる。初期データの算出では、除算処理も必要であるが、ほとんどが、単純な加算により求めることができる。大体100画素生成するのに、数千命令の演算が必要と言われている。

現状のグラフィックスシステムのほとんどは、上記パイプライン処理

の各部分を独立したハードウェアで実現しているため、各パイプライン間のデータ転送能力のバランスや、各部の演算性能のバランスが実効性能の向上のために重要となる。

### 3.2 グラフィックスハードウェアの構成

前節で述べたとおり、高速なグラフィックスを実現するためには、膨大な演算性能や、バス性能が必要である。このための基本的な性能向上策としては、プロセッサの高速化手法と同様、並列処理、パイプライン処理、などが用いられ、研究レベルでは、メモリ素子に論理を付加して高速化するなどのアイデアも提示されている。こうした動きから画像用専用メモリも生まれている<sup>4),5)</sup>。

製品化されているものを中心に、図-1のグラフィックスパイプラインのどの部分をハードウェア化しているかを分類分けすると、図-3のように大別される<sup>18)</sup>。

(1)は座標変換部以降をすべて、ハードウェア化したものである。座標演算処理部に高速なフローティング演算機構を有したデジタル信号処理プロセッサや、RISCプロセッサを並列に並べることで一度に複数の座標情報を処理する方式が一般的である<sup>6)~8)</sup>。このため、この構成は一般的に大規模な専用ハードウェアが必要で、三次元グラフィックス専用システム向けの構成と言える。性能レベルは、使用するフローティング処理部の性能に依存し、たとえば、100MIPS, 50MFLOPS程度の演算器を数個並べると、1Mポリゴン/秒以上の性能を実現できる。

(2)は、レンダリング部以下をハードウェア化するもので、図-2に示す処理を専用を実現できるハードを付加する方式である<sup>9),11),13),14)</sup>。図-2

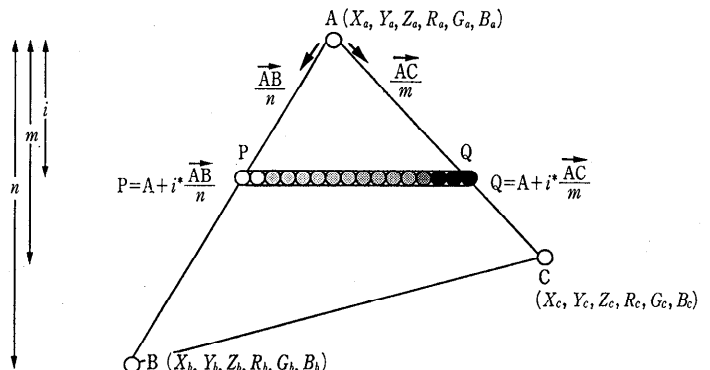


図-2 塗りつぶし処理におけるレンダリング処理

のすべての処理をハードウェア化するケースと一本のラスタ操作のみをハード化するケースの二種の方式が考えられている。CPU は、座標変換などの処理に専念できるため、前者のハードウェアを付加することで、100 MIPS 程度のプロセッサで、約 200—300k ポリゴン/s の性能を、後者のハードウェアで、その約半分程度の性能を実現することができる。図-4 は、後者のオペレーションの例で、ソースの画像を任意方向の任意長さのデスティネーションの画像に貼り付けることができるハードウェアである<sup>17)</sup>。二次元用のラスタオペレーションを機能向上させて、三次元にも対応できるようにしたので、拡張ラスタオペレーションと名付けた基本描画命令である。この機能を利用すると、以下のような各種のグラフィックス、イメージ処理のプリミティブコマンドとして使用できる。

直線展開の場合は、ソース画像として図-4(a)のように不連続線を定義し、デスティネーションとして任意方向の始終点を設定すると破線の描画に利用できる。

塗りつぶしの場合は、図-4(b)のようにソース画像として、なめらかに濃度の変化するラスタを定義し、デスティネーションとして水平方向の長さの違う始終点を定義して、繰り返し描画すると滑らかな台形塗りつぶしとして利用できる。

イメージ処理の場合は、図-4(c)に示すようにソース画像の部分ラスタを順次デスティネーションへ変換するように指定すると、画像のアフィン変換にも使用でき、高速なズームや、図に示すようなページをめくっているようにみせるようなオペレーションに使用できる。

こうしたハードウェアは、現状の LSI 技術では、一個ないし数個のチップに入れることができるため、比較的小きなハード追加で高性能化を実現することができる。

図-3 の(3)は、描画用の特別なハードは保有せず、単にディスプレイに表示するハードのみを有する方式である<sup>15)</sup>。性能はほかに比較すると低いですが、それでも、100 MIPS 程度の性能では、10k ポリゴン/秒程度の性能は実現できる。また、ソフトウェアを工夫し、処理を簡略化することで、さらに高速化を図ったり、あるいは、目的にあった処理に変更できるなどの自由度があり、今後、三次元グラフィックスの一般化にとまない、よく使用される構成とも考えられる。

#### 4. グラフィックス性能について

前節で、グラフィックス性能について、一秒間に塗りつぶし図形がどれくらい描画できるかの指標として M ポリゴン/秒という指標を示しながら説明した。こうした性能では、ユーザが一体どれくらいのことのできるのかが分からないため、最

No.	構成	システム構成	性能
1	座標変換/ レンダリング処理 のハード化	CPU、高速浮動小数点演算器、画像生成ハード プリント基板数枚	浮動小数点演算器の MIPS/MFLOPS、及び、CPU-専用ハード間パス性能に比例
2	レンダリング処理 のハード化	CPU、画像生成ハード LSI 数個	CPU の MIPS/MFLOPS に比例 MIPS 値の約500分の1~千分の1
3	すべて ソフトウェア処理	CPU、表示制御ハードのみ LSI 1個弱	CPU の MIPS/MFLOPS に比例 MIPS 値の約1万分の1

■：専用ハードウェア部

図-3 グラフィックスパイプラインとハードウェア化のレベル

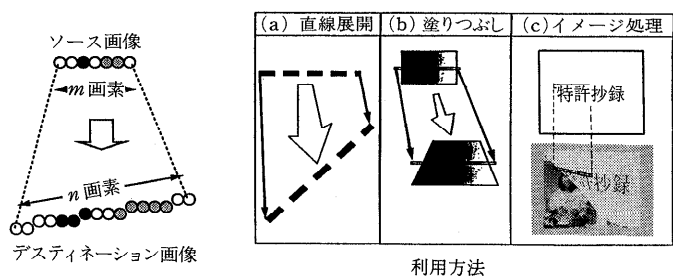


図-4 ラスタ演算用の拡張ラスタオペレーションとその利用方法

近では、描画性能のベンチマークで示そうとする動きが盛んになってきている。

性能評価指標としては、CPU 性能を測る指標が著名であるため、それと対比して、グラフィックス性能を測るベンチマーク群を表-1 に示す。グラフィックスの世界で、従来使用されていた、一秒間にどれくらいの線や、塗りつぶしができるかという M ベクトル/秒や、M ポリゴン/秒という性能は、CPU でいう加算命令や、メモリ参照命令がどれくらいの速度で動作するかという指標と同一で、単体コマンドの性能を示すものである。平均的な描画性能を示しているとは言い難いため、より実効性能に近い表現手法が必要になってきている。

表にある X BENCH, X Performance, Graphstone などが、その一つで、描画コマンドの加重平均などで性能を表す方法である。加重平均の重み付けに偏りがあるため、これを解消しようとするのが最近使用され始めた GPC ベンチマークで、実際の絵を描画する実行時間により性能を示すものである。これは、CPU の世界における SPECMARK ベンチマークプログラムに対応した、より応用レベルに近いベンチマークソフトと言える。

5つの絵(プリント板配線図, シャーシの構造図, 車のシリンダの塗りつぶし図形, 人の頭の塗りつぶし図形, シャトルのアニメーション)を例にとり、数百回の表示時間の逆数で性能を表現する。ほとんどの絵が、再描画を主体としていること、入力性能が含まれていないなどの問題はあがあるが、より応用サイドにたった性能指標で、今後とも拡張されて、より一般化されるものと思われる。

GPC ベンチマークで使用されている絵や、CAD の世界で実際に使用している絵について、ディスプレイ上に、どれくらいの数の描画コマンドが発行され、どのくらいのサイズで描画しているかをプロットしてみると、図-5 が得られる<sup>20)</sup>。

横軸が、一枚の絵のなかに含まれる描画コマンドの数を、縦軸が、そのときの描画する画素数の平均を示す。○印が実際の絵のデータで、双曲線は、CRT 画面の画素数を示す。グラフより、一つの絵を構成する図形数が増大し、絵が細かくなればなるほど、平均的な

塗りつぶし図形のサイズは小さくなり、一般的に次のような法則が成立していることが分かる。

$$\begin{aligned} & (\text{一枚の絵の図形数}) \times (\text{平均図形サイズ}) \\ & \approx \text{画面サイズ} \end{aligned}$$

一般的に、サイエンティフィックビジュアルライゼーションなどの世界では、図形データ量も 1M ポリゴンにおよぶものもあり、こうした世界では、一つの三角形あたり約 10-20 画素の塗りつぶしを実行する必要がある。一方、CAD の世界では、10k-20k ポリゴンのデータ量で、一つの三角形のサイズは、50-100 画素と大きくなる。ビジネスの分野で三次元が使用されるようになると、数百画素という比較的大きな図形を数千個描画するような応用が生まれてくるものと期待される。こうした応用ソフトの性質により、グラフィックスパイプラインの各段における負荷が変わってくるため、これらを考慮しながら、グラフィックス用ハードシステムの最適構成を実現することが重要になってくる。

表-1 性能評価指標の概要

	グラフィックス性能評価	CPU 性能評価
単体性能	ベクトル描画性能 塗りつぶし描画性能 文字描画性能	ロード命令性能 加算命令性能
加重平均性能	X BENCH X Performance Graphstone	ギブソンミックス ドライストーンベンチマーク リンパックベンチマーク
実効性能	GPC ベンチマーク	SPECMARK ベンチマーク

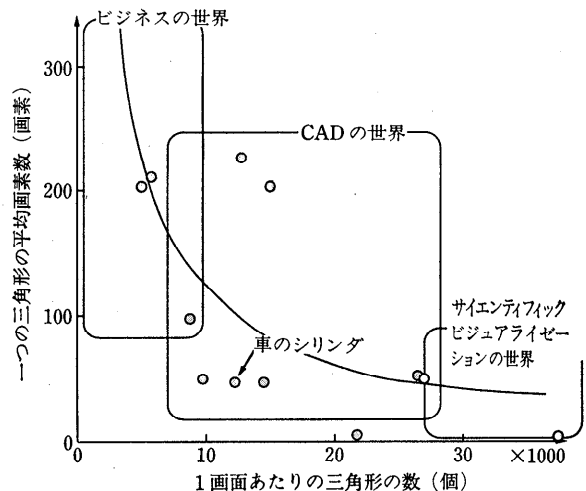


図-5 塗りつぶし中心の図形の描画コマンド数と描画サイズの関係

## 5. 応用ソフトウェアインタフェース

グラフィックスの応用展開を阻害している一因に、応用ソフトとのグラフィックスライブラリが、各社ごと、世代ごとにばらばらであるという要因があった。グラフィックスライブラリの標準化作業については、古くから委員会を中心に議論されてきたが、技術進展が激しかったため、定義されてもすぐに陳腐化され、なかなか決定的なものが出現しなかった。CORE, GKS, CGI などの標準化が進められたが、広く一般に使用されているとは言いがたいものがある。

ところが、最近になって、今後のグラフィックスインタフェースの標準化案としてよく話題に上っているものに PEX\* と OPEN-GL\*\* の二種類がある。旧来のインタフェースは、仕様書レベルでの標準化案で、インプリメンテーションは、メーカーに任されていたものが、今回の二者は、ともに実際のインプリメンテーションプログラムも提供されるため、今後使われていく可能性が高い。

PEX は MIT を中心とする X コンソーシアムで議論され、かつ実際のインプリメンテーションまで行われているもので、X ウィンドウの上に拡張されたものである<sup>21), 22)</sup>。ネットワーク上のプロトコルの応用インタフェースとしては、PHIGS+ と、より単純な PEXlib というライブラリが提供される。イーサネットワークを経由して、大量の三次元データは瞬時に転送できないため、あらかじめ送っておいたデータを再利用することで表示性能を高速化するケースでの編集系が豊富であるという特徴を有する。

一方、OPEN-GL は、シリコングラフィックス社がオープン化を推進しているライブラリで、コンピュータグラフィックスで必要となるテキスト処理（三次元面にイメージを貼り付ける処理）などに特徴を有する。さらに詳しい内容が必要な方は、たとえば、コンピュータネットワークの comp. graphics. opengl など議論されているため参照願いたい<sup>24)</sup>。

## 6. 今後の展開

以上述べてきたように、プロセッサ性能の向上

\* PEX は、MIT を中心に開発されているグラフィックスライブラリで、MIT の登録商標です。

\*\* OPEN-GL は、シリコングラフィックス社の登録商標です。

にともない、応用分野によっては、低価格なシステムでも三次元グラフィックスが扱える時代に入ってきた。ソフトウェア開発サイドからみても、1980年代の各社のワークステーションは、独自のグラフィックスインタフェースを提供していたが、1990年代は、標準の世界のなかで開発できるようになり、ハード非依存のシステム展開ができる素地が整いつつある。また、製品上も、ワークステーションに限らず、パソコンやゲーム機などにも展開されつつあり、基盤プラットフォームがそろってきている。

一方、三次元の活用方法についても、人工現実感や、三次元マンマシンインタフェース、さらには、リアルなゲームソフトなど、新しい分野が生まれつつある。世の中では、マルチメディアシステムに対する話題に事欠かないが、三次元グラフィックスは、マルチメディアをささえる一つのメディアとしても注目する必要がある。

こうした動きは、ちょうど、1980年代に、ビットマップディスプレイやマウスの一般化により、それまで、一部の人しか使用していなかった二次元グラフィックスや、イメージ処理が一般化した動きを彷彿させるものがある。

米国での関連学会の出席者や、日本国内でのイベントなども活性化してきており、多くの研究開発者がこの分野で活動を進めているため、市場要求に従って、ここ2-3年の間に大きく発展してくるものと思われる。

## 参 考 文 献

- 1) たとえば、Reddy, R.: User Interfaces in Giga-PC Environments, Plenary Talk at UIST '92 (1992.11).
- 2) James, F. and Andries, V.D.: Fundamentals of Interactive Computer Graphics, Addison-Wesley (1982).
- 3) Newman, W. and Sproull, R.: Principles of Interactive Computer Graphics, McGraw-Hill, 2nd edition (1979).
- 4) Fuchs, H. and Poulton, J.: Parallel Processing in PIXEL-PLANES, a VLSI Logic-Enhanced Memory for Raster Graphics, 1st International Conference on Computer Workstations, pp. 193-196 (1985).
- 5) Sproull, R.F. et al.: The 8 by 8 Display, Carnegie-Mellon University, CMU-CS-82-105 (1981.12).
- 6) Clark, J.H.: The Geometry Engine: A VLSI Geometry System for Graphics, Computer Graph-

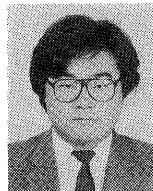
- ics, Vol. 16, No. 3 (1982).
- 7) Akeley, K.: High-Performance Polygon Rendering, Computer Graphics, Vol. 22, No. 4, pp. 239-246 (1988. 8).
  - 8) Priem, C. R.: Developing the GX Graphics Accelerator Architecture, IEEE MICRO, Vol. 10, No. 1, pp. 44-54 (1990. 2).
  - 9) Worley, W. M. S. and Worley JR. W. S.: Ardent's Fast Memory, VLSI System Design, Vol. 19, No. 8, pp. 50-59 (1988. 8).
  - 10) Levy, H. M.: VAXstation: A General-Purpose Raster Graphics Architecture, ACM Trans. on Graphics, Vol. 3, No. 1, pp. 70-83 (1984. 1).
  - 11) Rhoden, D. and Wilcox, C.: Hardware Acceleration for Window Systems, Computer Graphics, Vol. 23, No. 3, pp. 61-67 (1989. 7).
  - 12) Pang, A. T.: Line-Drawing Algorithms for Parallel Machines, IEEE Computer Graphics & Applications, pp. 54-59 (1990. 9).
  - 13) Voorhies, D. et al.: Virtual Graphics, Computer Graphics, Vol. 22, No. 4, pp. 247-253 (1988. 8).
  - 14) Voorhies, D. et al.: Reduced-Complexity Graphics, IEEE Computer Graphics & Applications, pp. 63-76 (1989. 7).
  - 15) Colwell, R. P.: A Display Architecture for Driving Two Different Bit Mapped Displays from One Frame Buffer, IEEE 1st International Conference on Computer Workstation, pp. 30-37 (1985. 11).
  - 16) 西沢, 大橋: 3次元グラフィックス処理プロセッサ, 電子情報通信学会誌, Vol. 72, No. 7 (1989).
  - 17) 藤田他: 拡張ラスタオペレーションによる3次元図形表示と画像操作, 情報処理, Vol. 31, No. 9, pp. 1334-1341 (Sep. 1990).
  - 18) Thayer, L. J.: A Overview of 3-D Graphics Workstation Architectures, GPC Quarterly Report, Vol. 2, No. 1, pp. 4-6 (1992).
  - 19) Dunwoody, J. C. and Linton, M. A.: Tracing Interactive 3D Graphics Programs, Computer Graphics, Vol. 24, No. 2 (1990. 3).
  - 20) 古賀他: ワークステーションのグラフィックスコマンドの分析と評価, 情報処理学会論文誌に投稿中.
  - 21) Rost, R. J. et al.: PEX: A Network-Transparent 3D Graphics System, IEEE Computer Graphics & Applications, pp. 14-26 (1989. 7).
  - 22) Sung, H. C. K. and Kubitz, W.: A Critical Evaluation of PEX, IEEE Computer Graphics & Applications, pp. 65-75 (1990. 11).
  - 23) Haletky, E. D.: Integration of GL with the X Window System, Xhibition 91.
  - 24) Akin, A.: Analysis of PEX 5.1 and Open GL 1.0, ftp で sig.com. の file sgi/OpenGL/analysis.ps より.

(平成4年11月30日受付)



福永 泰 (正会員)

昭和25年生。昭和50年京都大学大学院修士課程電気系工学研究科修了。同年(株)日立製作所入社。現在同社日立研究所主任研究員。計算機アーキテクチャ, グラフィックス, マン・マシンインタフェースの研究に従事。電子通信学会, ACM 各会員。



藤田 良

昭和33年生。昭和55年電気通信大学応用電子工学科卒業。昭和57年東北大学大学院修士課程修了。同年(株)日立製作所入社。現在日立研究所においてワークステーションの表示システムに関する研究に従事。電子通信学会会員。



古賀 和義 (正会員)

昭和33年生。昭和56年九州大学工学部電気工学科卒業。昭和58年同大学院工学研究科電子工学専攻修了。同年(株)日立製作所入社。現在日立研究所において表示システムに関する研究に従事。マン・マシンインタフェースに興味を持つ。電子通信学会, IEEE, ACM 各会員。