

## 分散環境を利用した並列動画生成

竹村真一 仲谷栄伸 砂原秀樹

電気通信大学 情報工学科

### 概要

科学技術計算のうち、特に結果の認識が困難である三次元科学技術計算の結果を可視化する場合、画像生成のための処理量が多く、アニメーションの作成に莫大な時間が必要となる。この問題を解決するために、複数のワークステーションを用いてアニメーション生成を高速に行なうシステムを実装した。ここでは実行環境として、近年一般的になりつつある分散環境を想定しており、スーパーコンピュータや画像生成専用コンピュータなどの特殊なハードウェアを必要としない。本稿では、システムの背景、概略および処理速度を中心とした性能について述べる。

## A Parallel Animation Production System in the Distributed Environment

Shin'ichi Takemura, Yoshinobu Nakatani, Hideki Sunahara

Department of Computer Science

The University of Electro-Communications

1-5-1 Chofugaoka, Chofu-shi, Tokyo 182, JAPAN

### Abstract

In scientific computer simulations, it is difficult to understand the results of them if they are threedimensional. It takes a great deal of time to make computation for an animation video pictures which visualize them. We implemented a high speed computer animation system on workstations in a distributed environment without special computers such as super computers or graphic super workstations. In this paper, the background, overview and performance in terms of execution speed, of this system are described.

## 1 はじめに

計算機を使用した大規模な科学技術計算は、直接実験・観察することのできない様々な現象を解明するのに大いに役立つとされ、近年特に注目されている。しかし、このような科学技術計算において、三次元空間中に分布する値や、三次元的な流れを示すような結果は人間が直接理解することが困難である。

この問題を解決するために、科学技術計算から得られたデータをさらに計算機で処理して視覚化する研究が近年盛んになりつつある。現代の科学技術計算にとって、このような可視化に関する研究は、数値技法の研究と共に非常に重要な位置を占める。

科学技術計算の中でも特に結果の認識が困難で可視化が必要とされるものとして、結果が三次元的な広がりを持つ場合が挙げられる。データが三次元的な広がりを持つ場合は、データに何らかの処理を加えて一般的な三次元図形を生成し、これを二次元に投影する場合と、特殊な手法を用いてデータを直接二次元に投影する場合が考えられる。前者の場合、三次元図形を生成する処理は表示対象によって様々であるが、二次元への投影には一般的な画像生成手法を用いることが可能である。このような画像生成手法については既に研究が盛んで、文献も多い。

本研究は、科学技術計算の結果から生成された三次元図形を二次元に投影する際に広く一般的に利用可能な画像生成システムの構築を目標としている。従って、科学技術計算を行なう多くの研究者がシステムを手軽に利用できるように、特殊なハードウェアをできる限り使用せず、一般的な計算機環境においての使用を前提とする。

特殊なハードウェアを使用せずに高速な処理を実現するため、複数の計算機がネットワークで相互に接続されている環境において画像生成プログラムを並列に実行する並列処理画像生成システムを実装し、実行時間を中心に評価を行なった。

## 2 画像生成システム

本研究は、科学技術計算の結果として得られる三次元空間に分布する値の可視化を行なう際に使用できる、特殊なハードウェアを必要としない画像

生成システムを提供することを目的としている。

### 2.1 システムに要求される課題

以下に三次元空間に分布する値などの可視化に使用することを前提にした画像生成システムに対する主な要求を示す。

- 科学技術計算では画面からより多くの情報が読みとれることが望まれるので、表示可能な要素数は数千～数万以上が必要である。
- 三次元空間に分布するデータが時間とともに変化していく場合は、画像<sup>1</sup>を連続して表示していくことによるアニメーションの作成が必要である。
- 娯楽向けの画像と違い、細部まではっきりと表示する必要がある。
- 三次元形状の内部まで観察するため、物体の属性として半透明が指定できなければならない。
- 生成された画像を表示対象に対する研究者の理解を助けるために使用する場合、生成された画像を確認した後、元のデータに変更を加えて再度画像生成処理を行なう作業が繰り返される。このため一連のアニメーション画像を短時間で生成する必要がある。

以上のように、本稿における画像生成システムは比較的高品位な画像の生成が必要であることに加え、アニメーションの作成を前提としている。

このような画像の例を図1に示す。この画像は外部から磁界を加えた時の磁性微粒子内部の磁化の様子を計算機シミュレーションを元に可視化したものである。

### 2.2 システムの概要

画像生成システムの概要を図2に示す。

画像生成部は科学技術計算の結果としてえられるデータ、視点、視野角、画面のピクセル数、必要なテクスチャデータなどの複数のデータをファイルから読んで画像を生成し、フレームバッファ<sup>2</sup>に出力する。フレームバッファに転送されたデータは

<sup>1</sup>本稿で単に「画像」とはアニメーション画像ではなく静止画像をさす。

<sup>2</sup>画像データを映像信号に変換して出力する装置の一般的な呼称。

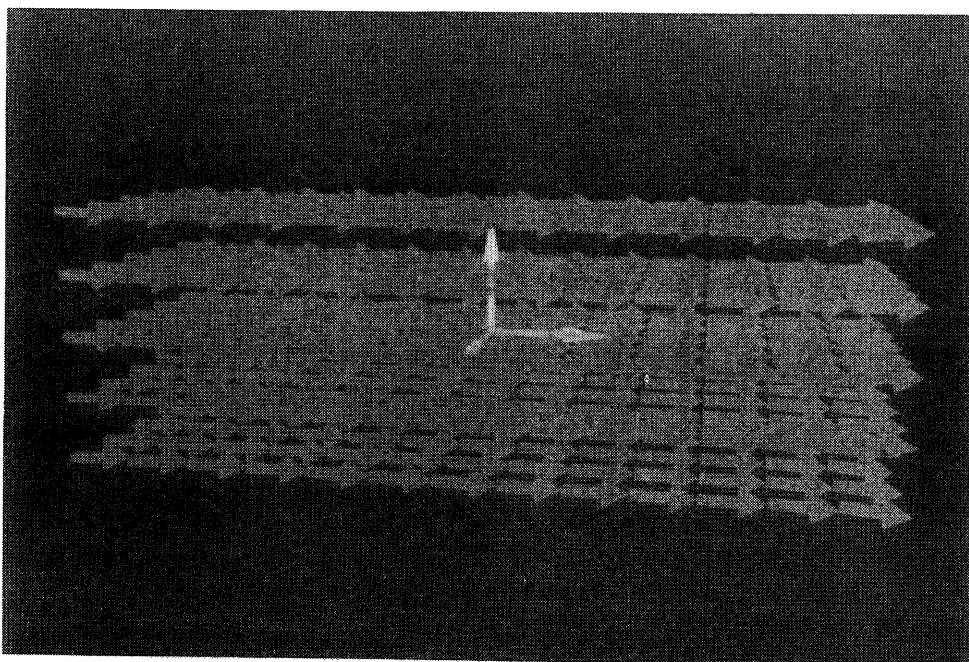


図 1: 科学技術計算の可視化の例

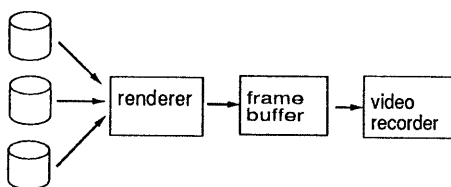


図 2: 画像生成システムの概要

CRT に表示可能な信号に変換され、画像として見ることが可能となるが、図 1 ような高品位の画像を実時間内<sup>3</sup>に計算してフレームバッファに転送することは、グラフィック専用機を使用しない限り不可能である。よって、計算された画像データは何らかの形で蓄えられなければならない。しかし、数十秒のアニメーションに相当する数千枚の画像データは数ギガバイトのデータ量を持っているので、これをそのままの形でディスクなどに記録することは現実的でない。ここではビデオ信号に変換された画像データを各画像毎にビデオレコーダに記録した後、元のデータは捨てる。利用者はビデオテープに記録された信号を再生することによって生成され

<sup>3</sup>人間の目に不自然な感じを与えないためには、毎秒 20 ～ 30 フレーム程度の画像を表示することが必要とされる。

た画像をアニメーションとして見ることができる。

現在実装されている画像生成部は隠れ面の除去のためにスキャンラインアルゴリズムに類似した手法を用いており、スムーズエッジ処理や半透明の表現が可能である。

## 2.3 システムの問題点

画像生成部は高品位な画像を出力できることを前提とした上でアルゴリズム的に高速化がはかられているが、通常のワークステーションでは 1 フレームの画像を生成するのに数分～数十分<sup>4</sup>の処理時間を必要とする。通常ビデオレコーダは毎秒 30 フレームの画像を再生するので、数十秒のアニメーションにも数百～数千フレームの画像が必要となり、かなりの処理時間<sup>5</sup>を必要とすることになる。2.1 節で述べたようにアニメーション画像を短時間で生成が必要であるのに、実際には数時間から

<sup>4</sup>図 1 の画像を Sun3/60(MC68020/20MHz) で計算した場合の処理時間は約 5 分である。

<sup>5</sup>上記の Sun3 で 30 秒（長めのテレビ CM と同じ時間）のアニメーションを計算した場合、およそ

$$5 \times 30 \times 30 = 4500 \text{ (75 時間)}$$

を必要とする。

数日も時間がかかってしまうことは問題である。

### 3 並列処理を用いた高速化

#### 3.1 今日の計算機環境

今日我々を取り巻く計算機環境は大きく変わり、数年前の大型機に匹敵する処理能力を備えたワークステーションを一人で一台占有できる環境が実現されつつある。

大学や研究所などには、それぞれに同程度の性能を持つワークステーションが多く設置されており、これらのワークステーション上では、データの整理や、プログラムの作成、文書処理、図面の作成などが行なわれている。

このような状況では、使用者が計算機に向かって作業を行なっている時間以外は、計算機がほとんど処理を行なっていないという場合が多く見られる。

これにともない利用者の少ない深夜には全く使用されていない計算機が数多く存在するようになった。このように現在我々の周囲には未使用の計算機資源が多く点在し、これらの余剰計算機資源はこれからも増え続けることが予想される。

画像生成ワークステーションは比較的高価でありあまり一般的でないため、これらの余剰計算機を活用することによって画像生成を効率的に行なうことは、科学技術計算の可視化に限らずコンピュータグラフィックスの一般化という面からも有効である。

本稿では、複数の汎用ワークステーションおよびフレームバッファとビデオレコーダが接続された1台のワークステーションが一般的なローカルエリアネットワークによって相互に接続された環境を前提としている。フレームバッファおよびビデオレコーダはコンピュータグラフィックスアニメーションの生成には必要不可欠な物であるが、あまり一般的でないため1組に限定した。

#### 3.2 並列化

現在の画像生成部は各画像間の相関関係を全く考慮せずに各画像を独立に計算している。アニメーションの生成において、各画像間の相関関係は処理の高速化に利用される場合があが、科学技術計算の可視化においてこのような高速化が可能かどうかはアニメーションの元となるシミュレーション

結果や可視化の際の表現方法に大きく依存すると考えられるので、ここでは各画像間の相関関係を無視することにする。このことから、アニメーションを構成する数百～数千枚の画像を単純に複数のワークステーションに分担させ処理すれば、処理速度の劇的な向上が可能であると考えられる。

今回実装したシステムは、それぞれのワークステーションにネットワークを用いて画像の元となるデータを転送し、レンダリングを実行させ、再びネットワークを用いて画像データを回収して順次ビデオレコーダに記録する方式をとっている。この方式はデータを一切ディスクに書き込まないので、ディスクを持たないワークステーションも利用可能であるという利点を持つ。

この並列画像生成システムを使用した場合、図3のように行なわれていた処理が、図4のように複数の計算機に分散して並列化され、 $n$  台使用時には、実行速度はほぼ  $n$  倍になると考えられる。

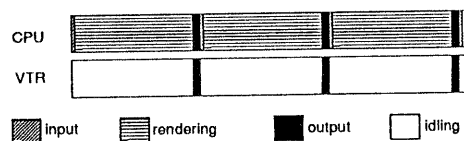


図 3: 通常の画像生成の様子

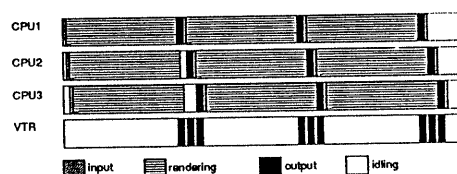


図 4: 並列分散処理における画像生成の様子

ネットワークの転送能力が高ければ転送は瞬時に終了するが、一般のローカルエリアネットワークでは中程度の転送速度を持つハードウェアが使用されており、転送速度を無視することはできない。図4の状態にさらに計算機の台数を追加して、十分な台数の計算機を使用すると、処理の様子は、図5のようになる。この状態では、処理時間は計算量に依存せず、各計算機間の転送時間で決定されるようになると予想される。

#### 3.3 並列計算モデル

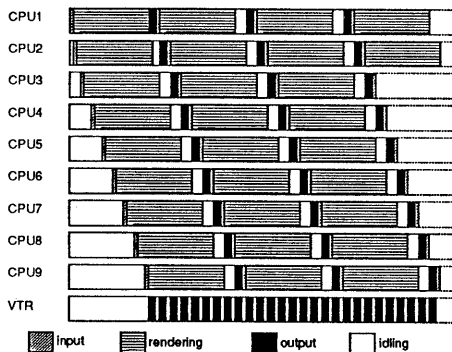


図 5: 十分な台数を使用した場合

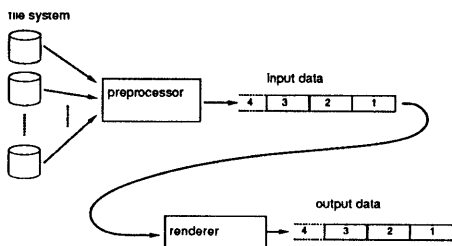


図 6: 計算機一台による処理のデータの流れ

計算機一台による処理のデータの流れを図 6 に、複数の計算機で処理した場合の流れを図 7 に示す。ただし、図 7 では図 6 中の preprocessor の部分を省略して描かれている。

preprocessor は科学技術計算の結果としてえられるデータ、視点、視野角、画面のピクセル数、必要なテクスチャデータなどの複数の入力ファイルを一つにまとめる処理を行なう。データ分配部はこのデータを適当に区切り、各ワークステーション上の画像生成部に転送する。それぞれの画像生成部は入力データに従って画像を生成する。データ収集部は、複数の画像生成部のうちいずれか一つと通信して画像データを受けとりビデオ機器に出力する。

preprocessor を設けることにより、データの流れを単純化し、通信のオーバーヘッドを軽減するとともに、ネットワークによる通信を行なうデータ分配・収集部と、画像生成に関与する部分を完全に分離している。

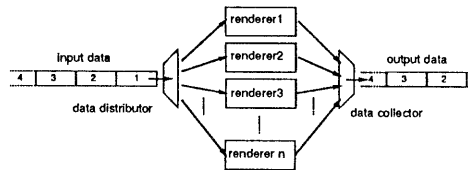


図 7: 並列処理におけるデータの流れ (部分)

### 3.4 並列画像生成システムの実装

ここでは、ネットワークを利用した並列処理画像生成システムを UNIX 上で実装した。ワークステーション間の通信には TCP を使用し、ワークステーション内のプロセス間通信にはパイプを使用した。

画像の元となるデータは通常数十 Kbyte 程度、通常のビデオ機器にあわせて解像度を設定すると 1 画面は 1Mbyte 程度であり、画像 1 枚当たりのデータ転送量は合計 1Mbyte 程度となる。秒間 100Kbyte 程度の転送が可能であると仮定すると、画像 1 枚当たり 10 秒で処理が終了することになり、1 時間で画像 360 枚が生成可能であると予想される。

並列処理画像生成システムの動作時の構成を図 8 に示す。

図中、画像生成の元となる科学技術計算データは予めディスクに蓄えられているとする。preprocessor に当たるプロセス arrow はこのデータを加工し、視点、解像度などの画像生成に必要な情報を付加する。

データの分配・収集を行なう parent は入力されたデータを 1 画面分毎に切り分け、各ワークステーション上の child に転送する。この時にソケットによるワークステーション間のネットワークを使用した通信路を使用する。

child は内部に適当な大きさのバッファを持っており、parent からのデータをこのバッファに格納する。1 画面分に相当するデータがバッファに蓄えられると、child は parent からのデータ受信を中断して、renderer に画像データをパイプを通して送出する。この時点で、parent は他の child へのデータ転送を開始している。

画像生成部である renderer はスキャンライン法によって画像を生成し、結果を child へパイプを通して出力する。出力は数分に渡って行なわれるが、このデータもまた child 内でバッファリングされる。

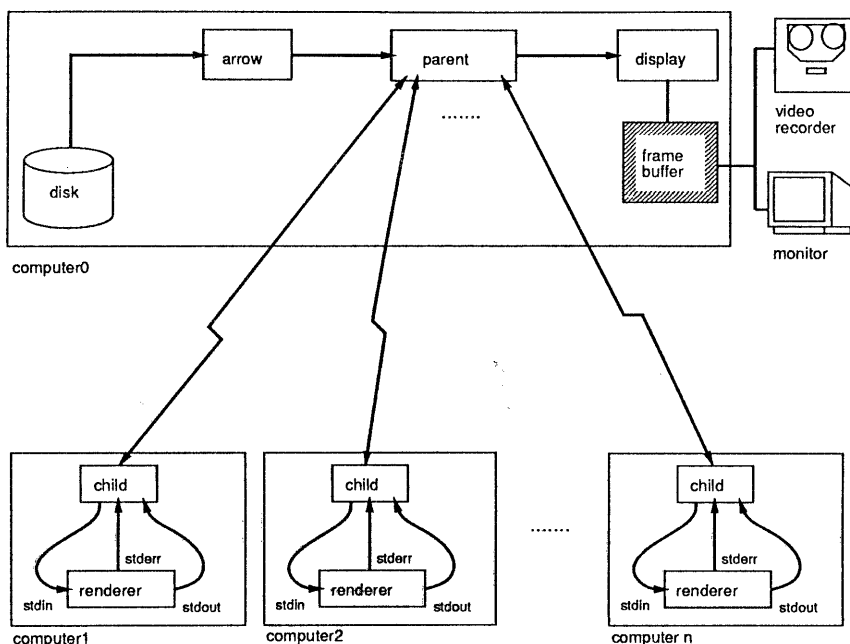


図 8: 実装した並列処理画像生成システム

1 画面分のデータがchild内に蓄えられると、childは計算の終了をparentに通知する。計算終了の通知を受けたparentは、childを内部の待ち行列に追加する。

parent は待ち行列の先頭の child にのみデータ転送を許可を与え、child 内のバッファから画像データを転送し出力する。parent は 1 画面分のデータ転送が終了すると、次に計算されるべき画像のデータを child に転送し、child を待ち行列の先頭から削除する。

以上で 1 画面分の処理が終了する。ここで重要なことは、データが child でバッファリングされることにより、1 画面分の入出力データの転送が数秒で終了することができる点である。各 child に対しての転送が数秒で終了するので、parent は次々に各 child に対して転送を行なうことができる。

画像生成部 renderer は単なるフィルタプログラムとして実装されており、図 6 および図 7 において全く同じプログラムを利用できる。

さらに parent の入出力データフォーマットは renderer と全く同じであり、相互に置換えが可能である。利用者は preprocessor および画像生成部を用意することにより様々な画像生成処理を並列に実

行できるが、preprocessor および画像生成部は通信処理とは全く無縁である。

## 4 性能評価

並列処理画像生成システムを用いて画像生成を行なった場合について、実行時間を測定した。測定は他の利用者が全くいない深夜に行ない、実行時間は全て実行開始から終了までの実際の時間である。

#### 4.1 实行環境

ネットワークを利用した並列処理画像生成システムの実行時間は、ネットワークの構成に大きく左右される。図9に実行時間の測定に使用したワークステーションと、それらを相互に接続するネットワークの構成を示す。

図中 brown と呼ばれる計算機はビデオ機器に接続されており、これに画像を記録できるようになっている。図8のプログラム arrow、parent、display は brown 上で動作させる。brown はイーサネットによってゲートウェイマシンである uecgw に接続されている。

実際に画像生成の計算に使用されるのは、計算機室の Sun3/60、blue0～indigo5 の 54 台<sup>6</sup> のうちのいずれかである。これらの Sun3 では child と renderer が一組ずつ動作する。これらの Sun3 及びゲートウェイマシンである pink はイーサネットで接続されている。pink は光ファイバケーブルを使用した回線で、brown 側のゲートウェイマシンである uecgw と接続されており、brown 上の parent はゲートウェイマシン 2 台を介して blue0 などの child と通信を行なう。

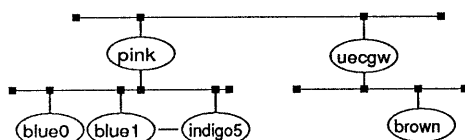


図 9: 実行ネットワーク環境

## 4.2 計算時間

前述の環境下において、並列処理画像生成システムで画像を生成した場合の実行時間を表 1 に示す。指定したデータから生成されたアニメーション画像は図 1 に類似するもので、99 枚の画像からなり、視点の移動や、構図の変化はない。ポリゴン数 8125、解像度 640 × 400、垂直方向 4 倍オーバーサンプリングを指定している。画像一枚に相当する入力データは約 36Kbyte、画像データは 760Kbyte である。

ネットワークを利用した並列化を行わずに計算機室の Sun3 を 1 台だけ使用して同様な画像を生成した場合と比較するため、blue0 において同様の画像を 20 枚連続して生成した結果、一枚当たりの処理時間は 337.86 秒であった。図中「所要時間」はプログラムの実行開始から終了までの実際の時間で、「時間/画像」は所要時間を生成した画像の枚数 (99) で除算したものである。この値は、実際に画像信号がビデオ機器に機器に出力される間隔とはほぼ等しい。「並列化による速度の向上」は、画像 1 枚を 1 台で計算した時の所要時間 337.86 秒を「時間/画像」の値で除算した結果である。ワークステーションの台数を増した場合の処理速度の増加はほぼ線形であり、30 台程度を使用した場合の実行速度は 1 台の場合に比べて約 25 倍という結果

<sup>6</sup> これらの Sun3 の処理能力はいずれも同程度である。

| 使用台数 | 所要時間<br>(秒) | 時間/画像<br>(秒) | 並列化による<br>速度の向上 |
|------|-------------|--------------|-----------------|
| 8    | 4534.720    | 45.805       | 7.376           |
| 14   | 2627.000    | 26.535       | 12.732          |
| 19   | 1914.680    | 19.340       | 17.469          |
| 24   | 1531.590    | 15.471       | 21.839          |
| 27   | 1412.080    | 14.263       | 23.687          |
| 29   | 1325.360    | 13.387       | 25.237          |
| 31   | 1240.700    | 12.532       | 26.959          |
| 34   | 1120.470    | 11.318       | 29.852          |
| 39   | 1071.950    | 10.828       | 31.203          |
| 44   | 1055.770    | 10.664       | 31.681          |
| 49   | 1051.400    | 10.620       | 31.813          |

表 1: 並列画像生成システムの実行時間

が出ている。しかし、ワークステーションの台数が 35 台以上では台数の増加に対する処理速度の増加があまり見られない。

図 10 に blue0 を 1 とした場合のシステム全体の処理速度と画像一枚当たりの各 child の待ち時間の平均を示す。台数の増加にともなって待ち時間が長くなり処理速度が横ばいになっていくと同時に待ち時間が増大していく。

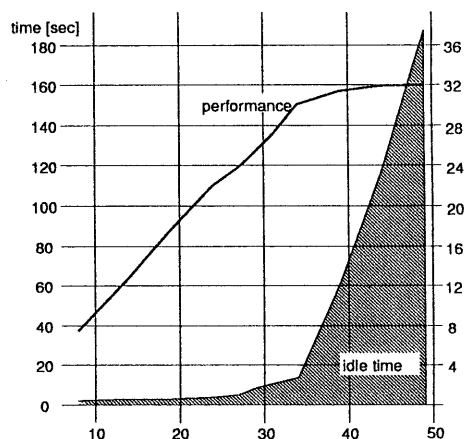


図 10: ワークステーションの台数とシステム全体の性能および待ち時間

## 4.3 ネットワーク負荷

システムのネットワークへの負荷を測定するために、図 9 の環境において稼働時のデータ転送量、データパケット数およびデータパケットのコリジョ

ン数を測定した。利用者の少ない深夜では、毎秒 90Kbyte ~ 100Kbyte の安定した転送を行うことが可能で、出力バケット数に占めるコリジョンの割合も 0.1%程度である。しかし利用者の多い昼間は、毎秒 50Kbyte ~ 90Kbyte と転送速度が安定せず、出力バケット数に占めるコリジョンの割合は約 1.5%程度である。昼間の測定では転送速度が低下したことにより、実行時間が深夜の 30%増しとなった。

#### 4.4 性能の限界

十分な台数のワークステーションを使用した場合の並列処理画像生成システムの性能は画像 1 フレームを構成するデータの転送時間によって大きく左右される。転送時間はデータ量と転送速度によって決定されるが、データ量は画像の性質によって決定されるので、ここでは議論しない。転送速度は child-parent-display の経路内でのデータのながれに複雑に左右される。イーサネットによるワークステーション間の通信の転送速度に加えて、ワークステーション内部のプロセス間通信にともなうプロセスの切替え、およびフレームバッファへの書き込みのオーバーヘッドは転送速度に影響を及ぼす。これらのオーバーヘッドはソフトウェアの変更によって軽減することが可能である。しかし、オーバーヘッドの軽減による転送速度の向上は、ネットワークへの負荷を増大させる。他の利用者がいない時間帯においてはイーサネットの通信容量に若干の余裕があるが、他の利用者が多い時間ではほぼ限界である。ソフトウェアを変更して、単純に転送速度を上げた場合、ネットワークを占有してしまう危険性がある。

## 5 おわりに

ワークステーションを相互に接続したネットワーク環境において、画像生成手法に関係なくアニメーションを高速に生成するシステムを実装し、ワークステーションを 1 台使用した場合と比較して数倍から数十倍の処理速度を実現した。

これまでは画像生成の専門家や業者によって行われていた画像の作成は、徐々に一般の研究者や計算機利用者の手によって行なわれるようになっていくと考えられる。一般的なワークステーション

を用いた並列処理画像生成システムはこのような場面で非常に有用である。

並列処理画像生成システムの処理能力は使用するワークステーションの台数および画像の複雑さとネットワークの通信速度で決定されるが、標準的なビデオ機器に記録する程度の画像を生成した場合、使用台数を増加させても通信速度の制限から毎分最大 5 ~ 6 枚の処理が限界である。ネットワーク全体のハードウェアをより性能の良いものに交換することにより転送速度の向上を望むことは経済的に難しいので、今後データ量を減少させることによって転送時間を短くする手法を用いて高速化をすべきである。また、処理速度がそれぞれ異なるワークステーションを使用した場合への対応などの課題も残されている。

## 謝辞

本研究の推進ならびに計算機の利用にあたって多大な御助力をいただいた電気通信大学情報工学科林信夫教授に心より感謝します。

## 参考文献

- [1] 西田 友是, 中前 英八郎: マルチスキニング法によるスムーズエッジ処理を施した三次元物体の陰影表示, 情報処理学会論文誌, Vol.25, No.5, pp.703-711
- [2] 山口富士夫: コンピュータディスプレイによる図形処理工学, 日刊工業新聞社, 東京 (1981)
- [3] Steven Harrington 著 郡山 あきら訳: アルゴリズムとプログラムによるコンピュータグラフィックス [II], マグロウヒル出版, 東京 (1988)
- [4] Sun Microsystems, Inc.: "SunOS Reference Manual", 1990.