

PC クラスタを用いた高精細大型立体ディスプレイシステムの開発

東京工業大学 大学院 情報理工学研究所

代表研究者 中嶋 正之 東京工業大学 教授
共同研究者 高橋 裕樹 東京工業大学 助手

目 的

近年、CAVE(CAVE Automatic Virtual Environment)等に代表される没入型ディスプレイを用いた研究が盛んに行われている。没入型ディスプレイによって作り出される映像空間では、高いインタラクティブ性とユーザの視野を覆い尽くす広大なスクリーンによって高い没入感を実現することが可能であり、様々な分野での応用が期待されている。

しかし、現在没入型ディスプレイを構築するためには非常に高性能な GWS(Graphics WorkStation)が必要となり、維持管理にも非常に大きなコストが必要となることが大きな問題となっている。

一方、近年 Beowulf に代表される、PC を並列化することによってスーパーコンピュータに匹敵する処理能力を実現する PC クラスタが広く普及しつつあるが、3 次元シーンのリアルタイムレンダリングへの応用はまだ少ない。

そこで本研究では、急速な高性能化を実現しつつある PC(Personal Computer)を複数台並列化して用いることで、様々な構成要求に柔軟に対応することの可能な高精細大型立体ディスプレイシステムに対する検討を行った。本研究では、大きく分けて次の 3 つの課題に取り組んだ。本報告では、(1)、(3)については概要を述べ、(2)について中心に述べる。

- (1) PC による立体映像出力とレンダリング負荷分散
- (2) PC クラスタを組み合わせた高精細大型立体ディスプレイ
- (3) 多様なスクリーン構成への対応

方 法

- (1) PC による立体映像出力とレンダリング負荷分散

本研究では、既存の PC において立体映像出力を可能にし、並列化された PC におけるレンダリング負荷分散を実現するために図 1 に示すステレオ映像同期ボードの提案を行った。本ボードを搭載した PC は専用の同期信号回線によって接続され、バッファリングした映像を同期させつつ出力することで時分割方式による立体映像表示を実現する。また、局所的なレンダリング負荷を分散させるとともに、ユーザの注目領域等に対するより高速かつ高精細なレンダリング処理を実現することも可能となる。

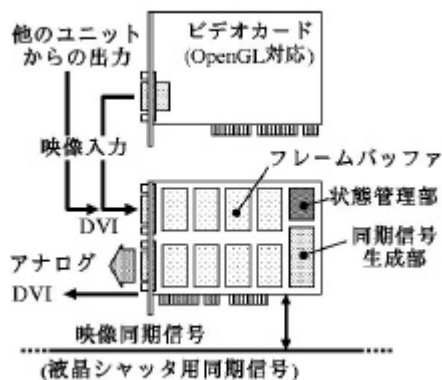


図 1. ステレオ映像同期ボード

(2) PC クラスタを組み合わせた高精細大型立体ディスプレイ

ステレオ映像同期ボードを搭載した PC を並列化することで図2に示すような高精細大型ステレオディスプレイの実現が可能である。また、PC クラスタを高速・高帯域ネットワークによって接続することで多彩な映像表現を可能にするための各種実時間演算処理の処理能力の検討を行った。具体的には、分散処理による画像のレンダリングの性能評価および衝突判定処理の分散アルゴリズムの検討を行った。さらに、ネットワーク帯域とレンダリング対象の変化による並列レンダリング性能について検討を行った。

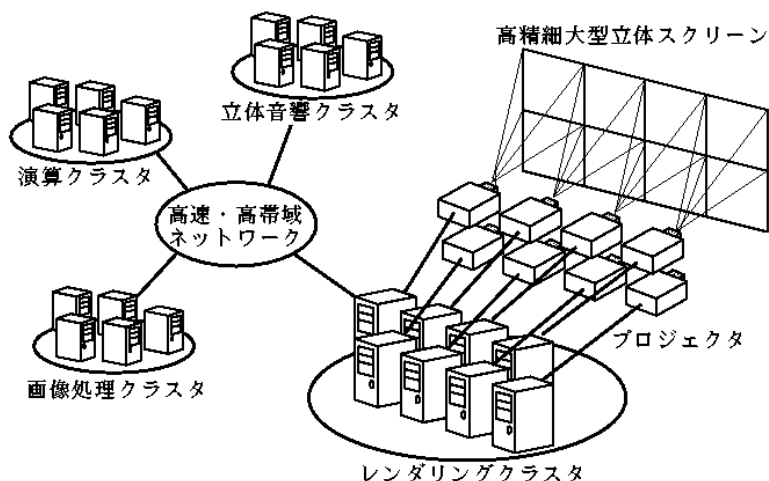


図2 システム構成

PC クラスタを CG レンダリングに使用するために、さまざまなシステムが提案され、それらに関する研究がなされている[1][2][3]。

また、そのシステムに関してまとめた研究[4]があり、その研究では、システムに分別に関して、アプリケーション分散型の ASE(Application-level Synchronized Execution)や、アプリケーション共有型の GLR(GL-DLL Replacement)があるとされている。従来、レンダリングクラスタシステムにおいては、アプリケーションレベルにおいてプログラマが個別に並列化を行なう ASE(Application-level Synchronized Execution)形式が多く用いられてきた。しかし近年、並列化に対応した graphics API への置き換えを行う GLR(GL-DLL Replacement)形式が提案されている。

本研究では VR システムとしての、マルチディスプレイシステムを構築する。そのため、レンダリングクラスタにおいて複雑なコンテンツをリアルタイムで描画することが必要がある。従って、GLR 形式を基にシステムを構築することで、リアルタイムレンダリングのために、高い描画速度を実現する。また、ASE 形式のデメリットである、並列化のためのプログラミングの手間を省くように、設計した。

GLR 形式の描画速度を上げることは、困難であることから、今回は GLR 形式において描画速度を上げる方向では無く、ASE 形式のシステムを基本に、システムを構築をする。

ASE 形式でのシステム構築において困難であるのが、通信するためのプログラムなどを構築することである。ASE 形式のシステムを構築するためには、OpenGL などのプログラムの中に、サーバプログラムを組み込まなくてはならない。従って、本研究でのシステム構築は、OpenGL のプログラミングスキルを持つ人であれば、簡単に画像を並列化して複数台の PC 上にて、インタラクティブな動作ができるようなシステムの構築を目指す。

本システムでは、並列化の作業を減らし、OpenGL をベースにプログラムしながら、少しのコマンドを追加、及び、変更をすることによって並列化を可能なものにする。

システムの概要に関して説明する。システムは、図 3 のように、ASE 形式を基に構築した。config ファイルを作り、そのファイルに IP アドレス、描画位置、解像度などの値を記述しておき、それをもとに、いくつかシステム側で用意した関数の発行によって、描画分割をする。プログラミング時には画像の分割を考えなくても良いように関数を設定した。また、インタラクティブ性を持たせるためには、外部からの入力に対して、値を変更しなければならない。そのために、変更すべき値に関しては、システムで用意した型で宣言することによって、その変数を各 PC 間で共有変数として、利用できる。したがって、プログラマが通信に関しての意識をしなくて済むような設計になっている。

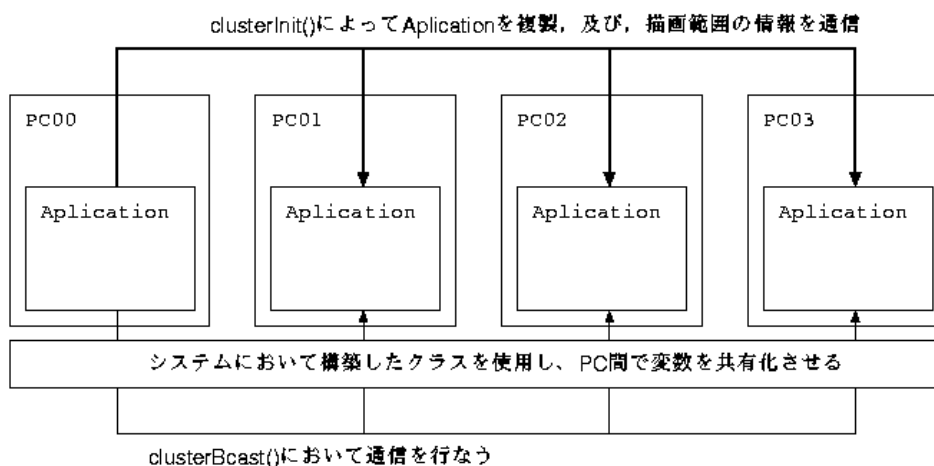


図 3 構築するシステムの概要図

また、今回のシステムでは、スクリーンが平面のものだけでなく、図 4 の(a)に示すスクリーンの角度がついてるものや、(b)に示す CAVE システムのように周りを囲んでいるものに対して、その角度を config ファイルに記述することにより、傾いたスクリーンへも投影できるようになっている。ただし、スクリーンの面は視線方向に対して垂直である必要がある。

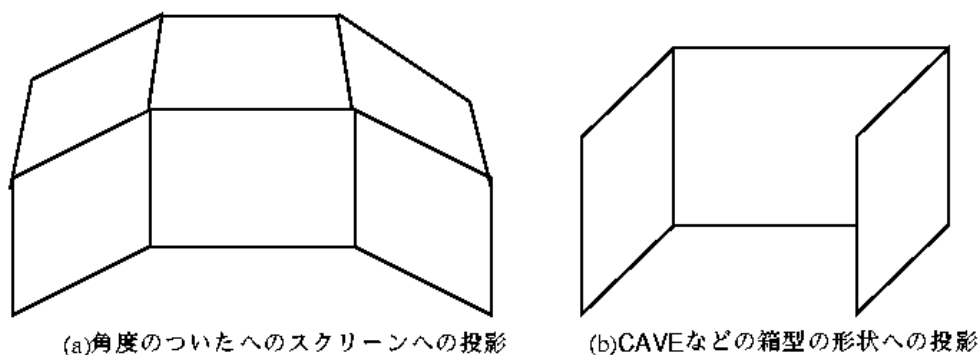


図 4 スクリーンの形状

本システムにおいて、構築したシステムでは、表 1 に示した関数を OpenGL のプログラムに組み込むだけで、ASE 形式の並列化をすることが可能である。また、clusterBcast 以外は OpenGL の関数と置き換えるだけであるから、特に並列化に関して考えること無しに変更をすることができる。

表 1 関数

function	備考
clusterInit	初期化の処理
clusterInitWindowSize	window サイズを決める
clusterSwapBuffer	描画の同期処理
clusterexit	プログラムの終了
clusterBcast	共有変数の通信
clusterFrustum	glFrustum と同様
clusterPerspective	glPerspective と同様
clusterLookAt	gluLookAt と同様

(3) 多様なスクリーン構成への対応

高精細大型ディスプレイを図 5 に示すように任意に配置することで、多様な映像空間を構築することが可能になる。また、用途に応じて容易にシステムを拡張することも可能であり、高いスケーラビリティを実現する。本研究では、現在 3 面のスクリーンを配置したシステムのプロトタイプを作成を行った。本研究は、現在実験進行中であり、まだ、外部での報告は行っていないが、今後学会での発表を予定している。

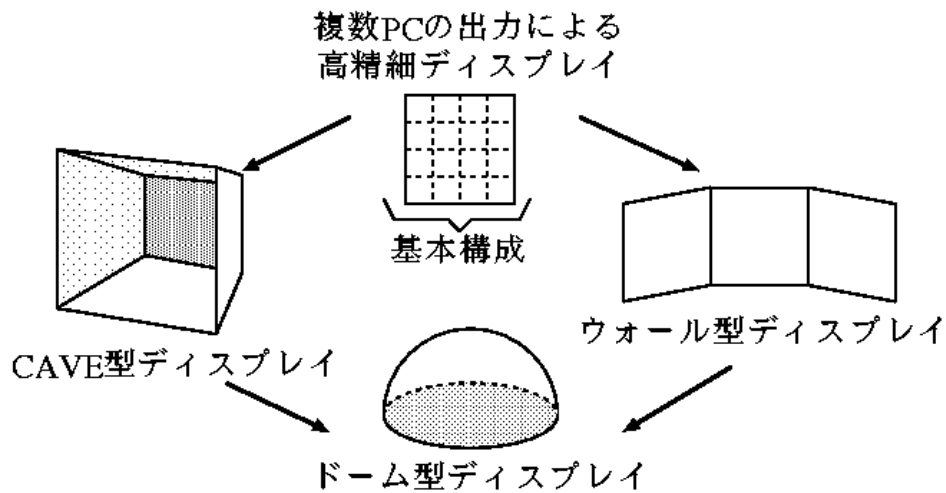


図 5 多様なスクリーン構成

結 果

作成したシステムを利用して、図 6 や図 7 のようなレンダリングクラスタ用のプログラムを作成した。

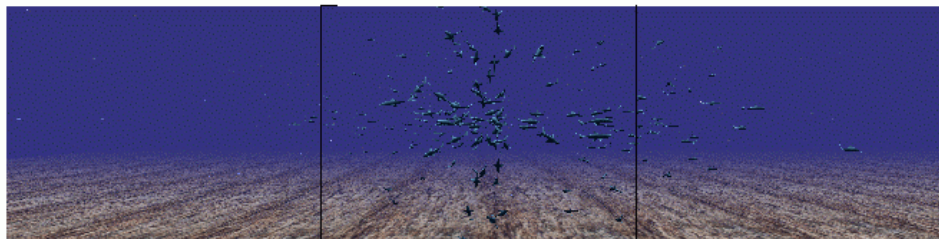


図 6 Atlantis

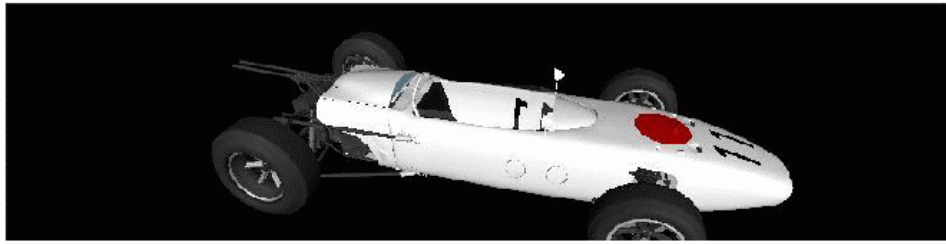


図 7 wr12view RA232.wrl

図 6 に示す Atlantis に関しては、海の中を移動するようにして、画像を動かしても、画像間のずれというものは、全く無く、スムーズに移動ができる。図 7 に示す wrk2view は VRML データの viewer であるが、この他にもさまざまな VRML データを読み込むことができ、視点を動かしてもその画像間のずれもなく、正確に同期が取れていた。

以上のように、本システムを利用することにより、ASE 形式において、プログラムが複雑になりがちなものを、いくつかの関数を定義し直し、変数を共有させることにより、並列化のプログラムを最小限の労力で構築することができるようになった。

(1)提案システムのリアルタイムレンダリング性能の評価

ここでは、本システムのレンダリングシステムに関して、描画性能の評価を行なう。前節で構築したシステムを用いて ASE 形式と GLR 形式の描画速度に関する実験を行う。GLR 形式には、WireGL を用いた。そこで両方のシステムの間には、ネットワーク帯域の違いにより、大きな性能の違いが、あると考え、ネットワークの帯域毎に、実験を行い、描画速度との関係を調べた。

本実験で使用した PC クラスタの構成は ASE 形式においても GLR 形式においても、図 8 のようにクライアントサーバ用の PC を 1 台、レンダリングサーバ用の PC を 3 台用意し、画面を横に 3 つならべて映像を作った。これらの各 PC のスペックは図 9 に示すとおりであり、10Mbps, 100Mbps, 1000Mbps, Myrinet と 4 種類のネットワークを使用することができる。

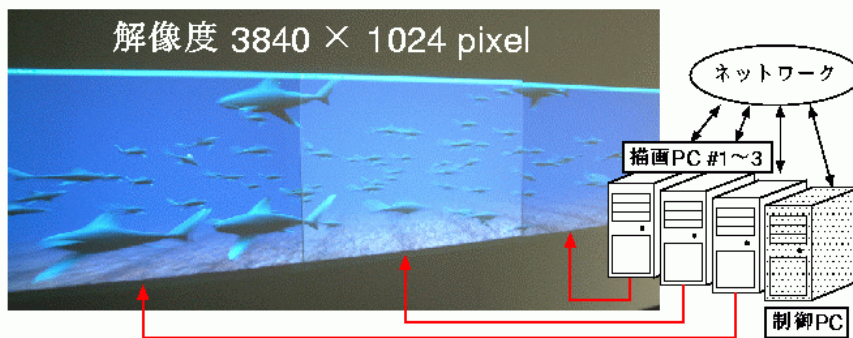


図 8 システム構成

```

CPU:Pentium4 1.8GHz
RAM:512MBytes
GrspicBoard:ELSA GRADIA920 Geforce3
Network:10Mbps,100Mbps,1000Mbps,Myrinet
OS:Linux Debian woody
  
```

図 9 実験で使用する PC のスペック

本実験ではクラスタレンダリングシステムの性能を評価するために、ネットワークの帯域を変化させ、帯域ごとに描画速度を測定することにより、ネットワークへの負荷を調べる。また、GLE 形式におけるデータ転送量の変化による描画速度の変化を調べた。

ここでは、クラスタレンダリングシステムにおける描画速度に関する実験の結果を示す。始めに ASE 形式と GLR 形式における描画速度の比較を行ない。次に、GLR 形式におけるデータ転送量と描画速度の関係について調べる。

(2) ASE 形式と GLR 形式の描画速度の評価

始めに、ASE 形式と GLR 形式の描画速度を計測した。その結果を表 2 に示す。

表 2 ASE 形式と GLR 形式の描画速度 (fps)

Application	ASE			GLR		
	A	B	C	A	B	C
10Mbps	81	46	6	0.5	0.8	8
100Mbps	81	46	6	6	8	14
1000Mbps	81	46	6	20	22	14
Myrinet	81	46	6	36	--	14

表 2 の PC とは 1 台で本実験で使用するコンテンツを実行したときの描画速度である。このときの描画速度が、PC の値であり、これらのコンテンツは CPU に対する負荷は大きくはないため、今回用いたグラフィックボードの性能であると考えられる。ここでわかるように、ASE 形式はネットワーク帯域を変えても、描画速度に差の無いことから、ネットワークへの負荷が少ないことがわかる。逆に、GLR 形式はネットワーク帯域を大きくすることで描画速度が速くなっていることから、ネットワーク帯域への依存が、大きいことがわかる

(3) GLR 形式の転送量と描画速度に関する実験

この実験では、GLR 形式の通信量と描画速度の関係を、ネットワークの帯域毎に測定した。結果は以下のようになった。

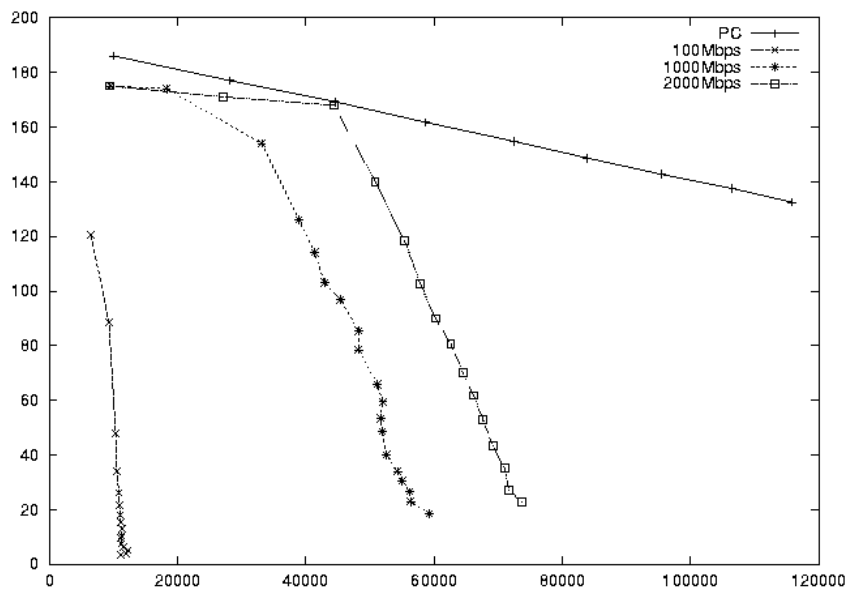


図 10 GLR 形式による通信量と描画速度

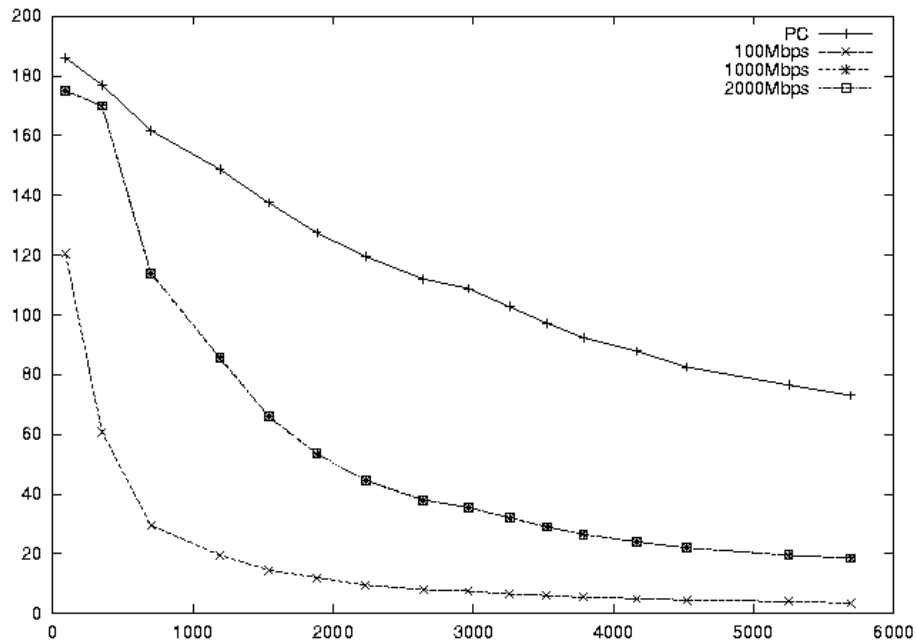


図 11 GLR 形式による描画の複雑さと描画速度

図 10 の 10MBytes/sec あたりでは 100Mbps の場合描画速度が急激に落ちてきている。通信量は、ポリゴンの増加、すなわち、コンテンツが複雑になるほど多くなる。したがって、ポリゴン数が多い 40MBytes/sec 付近では急激に低下していくのがわかる。これは通信量がネットワークの帯域の制限に非常に近く、データの転送量が限られるために、1 秒あたりのフレーム数が減少していることを表している。

図 11 は、1 フレームあたりのポリゴン数を横軸にすることにより、横軸をコンテンツの複雑さとして、それぞれのネットワーク帯域による描画速度を表したものである。図 11 のように、明らかにネットワーク帯域による描画速度の違いが分かり、PC と比べて GLR 形式の描画速度が低いことがわかる。

このように、100Mbps では GLR 形式のシステムは実用的でないことがわかる。また、1Gbps, 2Gbps においては、簡単なコンテンツであれば、十分に機能するが、複雑なコンテンツに関しては、実用的な描画速度を実現することが困難である。

以上のような実験により、本システムは高い描画性能を持っていることがわかる。GLR 形式は、アプリケーションの汎用性が非常に高い反面、通信速度がボトルネックになりやすく、十分な描画速度を得ることが困難である。本システムのような ASE 形式は、汎用 PC やグラフィックボードの急速な進化を考慮に入れると、現在の状況においては、VR システムとして最も適したものと考えられる。

本研究では、この GLR 形式によるリアルタイムレンダリングシステムを用い、PC 間のネットワーク通信に着目した描画性能に関する検討を行った。PC クラスタにおいて、ネットワークによる通信がボトルネックとなる。そのため、ネットワークの帯域の違いによる通信データ量を調べ、ネットワークへの負荷の大きさを測り、ASE 形式と GLR 形式との描画性能の比較検討を行った。また、その結果を考察し、ASE 形式に基づいた、VR システムへの応用を考え、レンダリングシステムの構築を行なった。ASE 形式はレンダリングシステム用のプログラムを構築するために独自にサーバをプログラムに組み込むものが多いが、本研究では、ASE 形式においても、プログラムの負荷を最小限に押えられるように、レンダリングシステムのための OpenGL ライクの並列化ライブラリを構築した。描画同期のために、各

PC 間に共有の変数を作れるようなクラスを定義し、それを用いることで、プログラムはクラスの定義において、各 PC 間に共通の変数を作ることが出来る。並列化プログラムを容易に作ることができるシステムを構築し、各種の VR システムとして利用できることを明らかにし、PC クラスタによって実現される莫大な演算処理能力を実時間処理に活かすことでより多彩な仮想映像空間を実現することが可能になる。

参考文献

- [1] Yuqun Chen, Han Chen, Douglas W. Clark, Zhiyan Liu, Grant Wallace and Kai Li.: "Software environments for cluster-based display systems", Proc. of IEEE/ACM International Symposium on Cluster Computing (2001).
- [2] R. Samanta, T. Funkhouser, K. Li and J. P. Singh: "Sort-First parallel rendering with a cluster of PCs", SIGGRAPH, Princeton University (2000).
- [3] R. Samanta, J. Zheng, T. Funhouser, K. Li and J. P. Singh: "Load balancing for multi-projector rendering system", SIGGRAPH/Wurographics Workshop on Graphics Hardware (1999).
- [4] Han Chen, Yuqun Chen, Adam Finkelstein, Thomas Funkhouser, Kai Li, Zhiyan Liu, Rudrajit Samanta and Grant Wallace: "Data distribution strategies for high-resolution displays", Preprint submitted to Elsevier Science (2001).

研究発表

1. 中嶋 正之、橋本 直己、川上 一郎: "高精細画像生成表示装置", 特願 2001-084244
2. 橋本 直己、中嶋 正之: "PC クラスタを用いた並列レンダリング手法の検討", 第 62 回情報処全大, 7V-2, (2001.3).
3. アグス バニアブディラー、橋本 直己、高橋 裕樹、中嶋 正之: "Linux-PC cluster と BMRT を用いた安価なオープンソース型 CG プロダクションシステム", 第 17 回 NICOGRAPH 論文コンテスト論文集, pp.9-14, (2001.11).
4. アグス バニアブディラー、橋本 直己、高橋 裕樹、中嶋 正之: "PC クラスタによる CG プロダクションシステム", 信学技報, Vol.101, No.201, pp.73-80, (2001.7).
5. 新保 達也、橋本 直己、高橋 裕樹、中嶋 正之: "PC クラスタによる 3 次元物体の衝突検出の並列化", 映情学技報, Vol.25, No.85, pp.13-18, (2001.12)
6. 新保 達也、高橋 裕樹、中嶋 正之: "PC クラスタによる衝突判定を用いた仮想空間の構築", 2001 信学ソ大, A-16-6, (2001.9).
7. 新保 達也、橋本 直己、高橋 裕樹、中嶋 正之: "並列化に適した衝突検出アルゴリズム", 2002 信学総大, A-16-24, (2002.3).
8. 薄井 武順、橋本 直己、高橋 裕樹、中嶋 正之: "PC クラスタのネットワーク帯域によるレンダリング性能の比較", 2002 信学総大, A-16-23, (2002.3).

連絡先

〒152-8552 目黒区大岡山 2-12-1 東京工業大学 大学院 情報理工学研究科