情 報 メ デ ィ ア 2－2
（1991. 7. 12）

# キャンパスネットワークとビジュアルコンピューテイング

山本強

北海道大学大型計算機センター

計算機環境インフラとして高速ネットワークが普及してきている。北海道大学においても北海道大学情報ネットワークシステム (HINES) の設置が進行中であり 1990 年から一部稼働に入っている。従来のコンピュータネットワークが端末吸収型あるいは計算機の遠隔使用が主要な目的であったが、TCP/IP あるいは OSI といった開放型のプロトコルにより異機種、ことなる OS の相互接続も可能になってきており、今後予想されるネットワークの速度・容量の増加を考慮すると２点間の論理的な接続を越えて広域情報バスとして高速ネットワークが認識される場面が増えてくると考えられる。本報告ではネットワークを高速情報バスとして用いた応用システムの例として Parallel Ray Tracer, HyperMap を紹介する。

# CAMPUS NETWORK AND VISUAL COMPUTING

Tsuyoshi Yamamoto

Hokkaido University Computing Center

N-11, W-5, Kitaku, Sapporo 060, Japan

High speed computer network is getting recognized as a new infrastructure for computing environment. At Hokkaido University, Hokkaido University Information Network System (HINES) is under construction and has been in service partly since 1990. As speed and capacity of computer network increase, it can be considered as a wide area information bus rather than point-to-point connector for connecting terminals and remote use of main frame computers, that are two major application of computer network used to be. In this paper, I introduce two application systems, Parallel Ray Tracer and HyperMap that use campus network as high speed information bus.

# 1 まえがき

コンピュータネットワーク、特に10Mbps以上の高速ネットワークの普及は最近10年の計算機環境における最も大きな変化の一つと言える。この情報バスが先進的計算機ユーザだけではなく電話と同一レベルで個人がアクセスできるようになるのにそう時間はかからないと断言できる。このような環境が実現されてその恩恵を受けるメデイアの一つが画像メデイアである。画像-視覚は計算機-人間を繋ぐ情報チャネルの中で最も帯域の広いものであり、そのインターフェースが改善されることはマン-マシンインターフェースの大幅な向上を意味する。そういう意味で高速ネットワークはユーザーインターフェースの重要なコンポーネントであると考えるべきである。

幸い北海道大学においては北海道大学情報ネットワークシステム (HINES) という名称のキャンパス LAN の設置が進行中であり、一部の機能は既に利用に供されている。HINES には公共サービスシステムとしての側面と情報ネットワークインフラとしての側面がある。キャンパス LAN は内線電話と比較されることがある。電話は通話サービス、あるいは通話行為を経由したサービスを提供するのであって電話線を直接さわろうとすると叱られるが、キャンパス LAN は定められたルールを守れば公共サービス (Mail, 仮想端末) 以外に自分で定義したアプリケーションプロトコルを流せるところが異なる。

キャンパス LAN には多くの受動端末、いわゆるターミナルサーバが接続されるがそれ以外に100台以上の Unix ワークステーションが接続されている。キャンパス LAN が CPU 間のバスとなっていると考えると 10Mbps という低速ではあるが100台以上の Unix ワークステーションが接続されている環境は一種の並列計算機と見ることもできる。この計算パワーを積極的に利用しようという試みは従来からあり、分散環境を目指す OS ではこの種の並列計算プロトコルを持つものがある。しかしキャンパス LAN の特殊性から既存の RPC プロトコルは容易に破綻してしまう。

開放型のキャンパス LAN には以下のような常識がある。

1. 機器の接続･切り放しが日常的に行なわれる
2. 接続する機器を指定できない
3. 使用できる通信容量が不定である

キャンパス LAN を用いたアプリケーションシステムおよびプロトコルはこれらの常識の上に成り立たたねばならない。

本報告ではキャンパス LAN をフィールドとして実験的に作成した2つのアプリケーションシステム, Parallel Ray Tracer と HyperMap を紹介し、ビジュアルコンピューテイングを例としてキャンパス LAN の応用と可能性を探る。

# 2 Parallel Ray Tracer

Parallel Ray Tracer はネットワーク上に分散配置された Unix ワークステーショングループを用いる並列画像生成システムである。単純に高速に画像を生成するならばいわゆるグラフィクスワークステーションを購入する事で解決できる問題であるが、ここで対象としている画像は光線追跡アルゴリズム [3][2] によるフォトリアリステイックな画像である。生成画像のクオリティをある程度犠牲にすれば高速の画像生成アルゴリズムも存在する。しかし画像のクオリテイ自体もユーザーインターフェースの重要なパラメータであり、可能な限り高速に高クオリティの画像が生成できることのメリットは大きい。光線追跡アルゴリズムは潜在的に並列度が高いことが知られており、出口 [1] らによる専用並列計算機の試みもある。キャンパス LAN 上に分散するワークステーション群を用いて並列計算を行なう事により専用計算機以上の実行速度を得る事が Parallel Ray Tracer の目的である。

Parallel Ray Tracer の目指す画像生成環境はネットワーク上のどのワークステーションからでも並列光線追跡のメリットが享受でき、キャンパスネットワーク独特の不安定性に対するロバスト性を持つものである。キャンパス LAN 上のマシンは全て自分の管理下にあるわけではないのでその機種も様々であり、さらに常時使えるとは限らないという非決定性がある。できるならCPU 負荷が一定値以下の場合のみ計算に加わる方式が望ましい。こういったヘテロジニアスな環境でかつ非決定性をもつ場合に使用できる分散計算プロトコルは見あたらない。Parallel Ray Tracer では RPC に相当するプロトコルのレベルから開発している。

## 2.1 Parallel Ray Tracer の実行モデルとプロトコル

光線追跡法は各画素の評価が独立に行えるため究極的には画素数、つまり $10^6$ のオーダの並列度が達成できる。しかし Parallel Ray Tracer が想定している CPU 数は4~32 程度であり、計算の粒度は画素と言うよりはブロックあるいは走査線である。計算粒度が大きくできると通信オーバーヘッドが減少するためキャンパス LAN のような容量の小さいネットワークでも負荷をかけないで済む。図 1は Parallel Ray Tracer の計算分割のメカニズムを示している。マスタプロセッサは数本の走査線を 1 単位として生成パラメータとともにスレーブプロセッサに渡し、その計算の結果生成された Pixel 列をマスタプロセッサに返す。図1においてスレーブプロセッサの数およびその識別名が実行時まで判明しない点がプロトコル設計上の問題となる。

Parallel Ray Tracer の RPC プロトコルは UDP(Universal Datagram Protocol) を用いて実現されている。多くのアプリケーションではより信頼性の高い TCP/IP プロトコルを用いるが TCP/IP はデータ送受の前に Point-to-point の接続を確認せねばならず、通信相手が不特定である場合には都合が悪い。UDP は Connection-

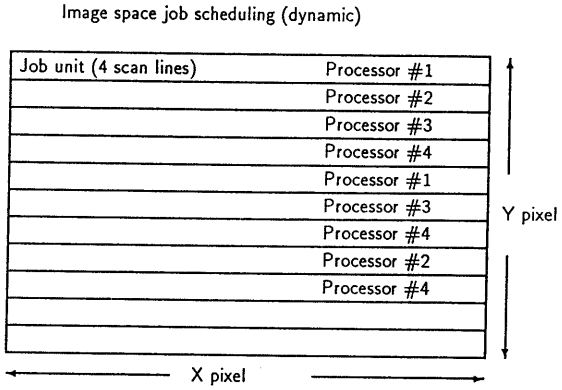


図 1: 光線追跡の分散化

less のプロトコルであるため不特定多数のホストとデータグラムの交換が可能であるが、配信確認が取れない欠点がある。配信確認はアプリケーションプロトコルのレベルで行なう必要がある。図 2は Parallel Ray Tracer でのマスタプロセッサとスレーブプロセッサの協調動作を示している。スレーブプロセッサ上のサーバプロセスはマスタプロセッサから rsh により起動される。各サーバプロセスは形状モデルのデータベースを読み込むがこれは rsh の標準入力のリダイレクションによる場合とあらかじめデータベースを各プロセッサに配布しておく方法が考えられる。rsh の標準入力から形状モデルデータを送る場合は初期化時にネットワークトラフィックが集中するため大規模な形状データを用いる場合には不利になる問題がある。今回に実験ではあらかじめデータを rcp により全プロセッサに配布している。

スレーブプロセッサ上のサーバプロセスはそれが正しく初期化された場合にあらかじめ定め

られたポート番号を用いてマスタプロセッサに Job request を送る。マスタプロセッサは全てのスレーブプロセッサ上でサーバプロセスを走らせた後でスレーブプロセッサとの対話ループに入る。マスタプロセッサ上のクライアントプログラムはスレーブからの Job request に対して Job parameter を返し、その結果としてフレームバッファ更新メッセージを受け取る動作を反復する。Parallel Ray Tracer ではクライアントが能動的に Job 割当を行なうのではなく受動的にサーバからの Job request を待つ形式のプロトコルとなっている所に特徴がある。スレーブプロセッサが稼働状態でない、あるいは高負荷状態である場合、rsh で起動しようとするサーバプロセスが全て起動に成功するとは限らない。しかしこのようなプロトコルでは失敗した場合でもスレーブプロセッサから Job request が出ないだけの事であり、有効なプロセッサ台数が減るが画像生成動作は遂行される。このように Parallel Ray Tracer では UDP による Connection-less のトポロジとマスタプロセッサがパッシブな形で Job の割当を行なう方式によって非決定性のあるサブプロセッサ群と協調動作を行なっている。

## 2.2 Parallel Ray Tracer の評価

キャンパス LAN 上に分散している Unix ワークステーションにアカウントを作成し Parallel Ray Tracese のサーバ及びクライアントを配布する。その後一台をマスタとしてクライアントプログラムを起動する。クライアントプログラムはその最初の動作としてリモートホストのリストに記述されたマシン上にサーバを rsh を用いて走らせる。Parallel Ray Tracer の並列計算メカニズムが有効に機能しているか否かはリモートホストの台数の増加と実行時間の関係で評価されるべきである。表 1 に代表的な作画にいついて CPU 台数と実行時間の実測値を示す。表 1 において $T1$ は Parallel Ray Tracer をコマンドとして起動して最初の画像を得るまでの時間であり、スレーブプロセッサの起動およびユーザーインターフェースの起動の時間を含んでいる。$T2$ は 2 回目以降の作画で初期化の時間を含まないため並列度が高く、台数効果が顕緒に現われている。1 台のプロセッサの時の $T1, T2$ の差、約 10 秒が初期化に要する時間と考えられる。

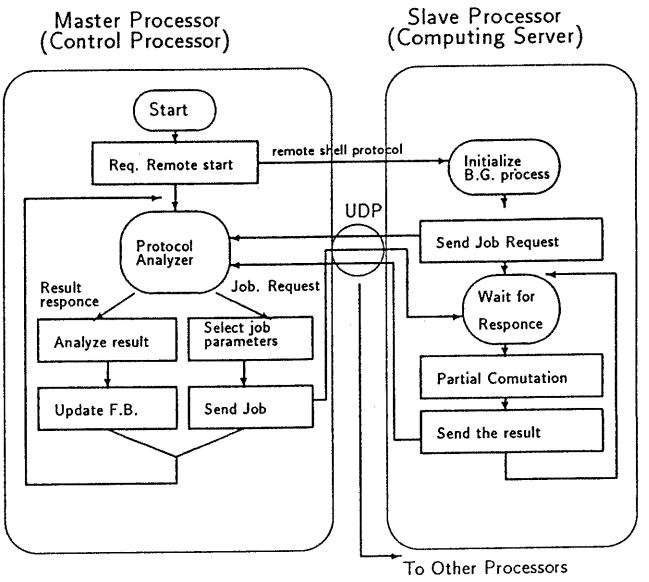


図 2: マスタプロセッサとスレーブプロセッサの協調動作

図 3 は $T1, T2$ を CPU 台数を横軸にプロットしたものである。$T2$ に関しては 12 台程度ではまだ飽和していない。図 4 は 12 台の SPARCStation1+ による 640x400 画素の画像生成過程を 5 秒間隔で記録した結果である。この場合、約 15 秒で計算が完了している。モデルデータとして約 2000 個の原子からなる結晶構造のデータを用いている。各スレーブプロセッサが非同期にフレームバッファを更新するため並列動作している CPU の数だけ走査線が同時更新されている様子がわかる。

今回は 12 台の SparcStation1+ に一時的なアカウントを作成して計測したが光線追跡の場合にはまだ計算時間が支配的でありネットワークトラフィックには余裕があることが確認できた。

表 1: CPU 台数 vs. 実行時間

| N (CPU 台数) | $T1$ | $T2$ |
|---|---|---|
| 1 | 103.4 | 92.6 |
| 2 | 62.2 | 51.7 |
| 4 | 35.1 | 24.5 |
| 8 | 23.3 | 13.4 |
| 12 | 20.2 | 9.5 |

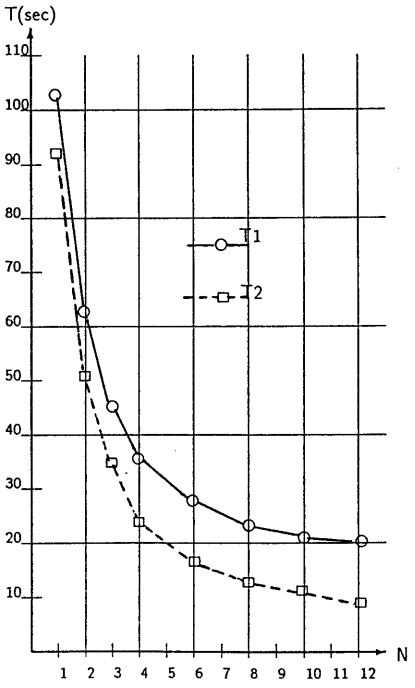


図 3: Parallel Ray Tracer の台数効果

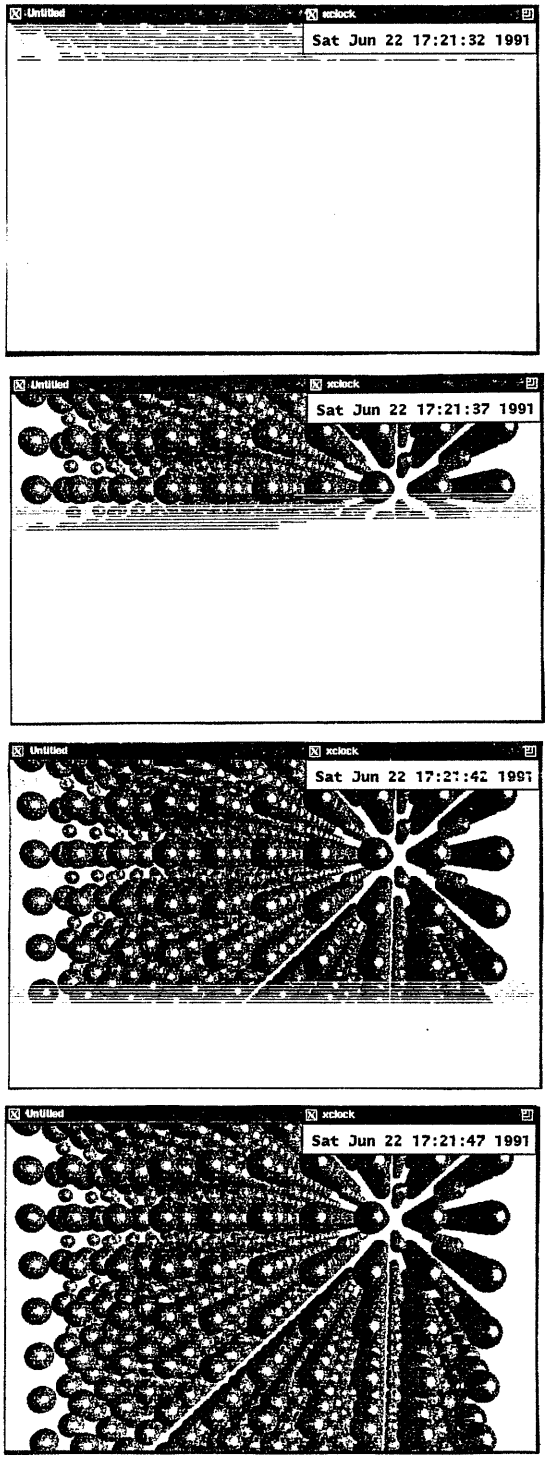


図 4: Parallel Ray Tracer の画像生成過程

## 3 HyperMap

HyperMap はネットワーク上に分散格納された動画像メデイアを汎用 EWS 環境から対話的に検索し、再生およびシナリオ作成を行なうシステムである。HyperMap においてキャンパス LAN は低レートアニメーション伝送バスとして使用される。低レートとはいえ数千コマからなるアニメーションデータは 100MB のオーダとなるため一台のサーバに複数のアニメーションデータを格納するのは困難になる。HyperMap ではキャンパス LAN 上に分散されたアニメーションサーバを X ウインドゥ上に作成されたユーザーインターフェースからアクセスすることを想定している。

(1) HyperMap のコンセプト

HyperMap 開発の基本となるコンセプトは多数の動画像セグメントを一般のユーザーがデータ構造を意識することなくアクセスするツールを提供することである。現在ではハンデイ型 VCR の普及やコンピュータグラフィクスの一般化で動画像データを入手することは容易になってきているがそれを利用するためのツールは一般化していない。その結果、多量の動画像データが作られたはしから捨てられているのが現状である。こういった状況を改善するためにビデオデイスクや CD-ROM を用いたマルチメデイアデータベースやオーサリングシステムが開発されている。しかしこれらの実験システムはハードウェアに大きく依存しており、実際に普及するためには規格の標準化を待たねばならない。HyperMap は動画像メデイアを現状のキャンパスネットワークと汎用ユーザーインターフェースの上で取り扱うことの可能性を探る目的で開発された。

HyperMap では動画像を 2 次元の空間的な地図(トポロジマップ) と対応ずけることによってアクセスする。図 5 は HyperMap のユーザーインターフェースの方法を示しているが、マップの要素は一枚の画像が連想される節点 (Node) と任意の動画像セグメントが連想される枝 (Segment) のみの単純なものである。

動画像データベースは通常の Unix ファイル形式であり 1 フレーム当り 300×200 画素、1 画素当り 8bit で表現されている。画像はあらかじめ Dither 法で中間調を 8bit にエンコードしてある。現在のファイル形式では 1200 コマのアニメーションが約 70MB のファイルに格納されることになる。

HyperMap は動画像ファイルおよびトポトジーマップを指定するスクリプトファイルを引き数として起動される。起動画面の一例を図 6 に示す。図 6 の上部の 2 個のウインドウは 300×200 画素のアニメーション及び静止画ウインドウであり、1 画素当り 8bit でカラー画像を近似する。左下部の比較的大きなウインドゥはトポロジマップであり、マップとしてカラー画像を用いることができる。図 6 では北大キャンパス内の景観の VCR(VHSC テープ 20 分間) を 1 秒当り 1 コマで 1200 コマを動画像ファイルとしトポロジマップのベースには構内地図を用いた。VCR 撮影の順序とトポロジマップの作成順序は関係なく、有為なセグメントを対応する地図上の 2 地点にセグメントとして定義する。ユーザーとして使用する場合はマップ上で節点をキーとしてセグメントを指定することにより動画像が呼び出される。連続するセグメントを指定することにより撮影順と関係なく動画像セグメントが連続して再生されることになる。これは一種の編集操作あるいはシナリオ作成と見なせる。

HyperMap の開発ではなるべく汎用プロトコルを用い容易なインプリメントをめざしている。基本的なアニメーションデータのネットワーク伝送は NFS による分散ファイルにより行なった。NFS ではプログラミング上はファイルの分散を意識する必要がなく、ネットワークインターフェースの知識がなくともインプリメントできる。ユーザーインターフェースは X11 および Athena Widget を用いて作成されている。このような環境で 300×200 pixel のアニメーションを表示させた場合のリフレッシュレートの実測値が表 2 である。NFS がファイルの読み込み

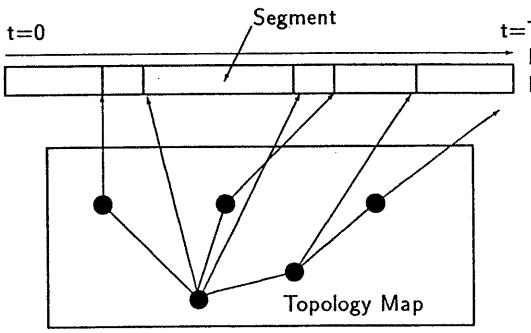


図 5: HyperMap のイメージ

表 2: ネットワーク分散ファイル経由のアニメーションリフレッシュレート

| Console CPU | $N1$ | $N2$ |
|---|---|---|
| SPARCStation1 | 4.5 | 13.1 |
| SPARCStation2 | 5.9 | 20.0 |

$N1$—キャッシュされていない場合のリフレッシュレート (frame/sec)
$N2$—キャッシュされた場合のリフレッシュレート (frame/sec)
(NFS サーバは SUN4/110)

に対して数 MB のキャッシュを持つため一旦アクセスした 100 コマ程度の動画セグメントは繰り返し再生される場合に X ウインドゥの描画速度程度まで向上する点である。SPARCStation2 では最高 20 フレーム/秒のほぼ完全なアニメーションに相当するリフレッシュレートが実現できている。

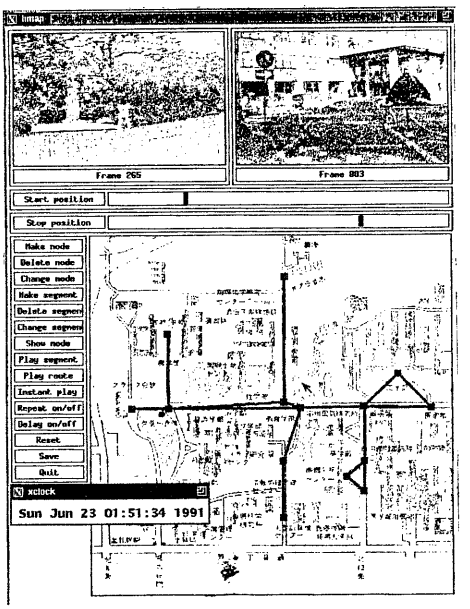


図 6: HyperMap の実行例

### 3.1 HyperMap の応用例

図 7は HyperMap を CG アニメーションのビュアーとして使用した例である。この例では MRICT により得られた 3 次元ボリュームデータをもとに多方向から見た画像をあらかじめ生成しておき、そのセグメントをトポロジマップを経由してアニメーション再生している。この例では半透明表示した頭部の画像を表示しているが、このような複雑な画像では動画像表示により内部構造が認識できる場合がある。この場合は頭部をいくつかのパスで巡回する形のアニメーションを 1000 コマ程度作成しておき、視点をトポロジマップの形式で指定している。

## 4 むすび

画像メデイアを取り扱うインフラ構造としてキャンパス LAN を取りあげ、その上で試験的に開発した二つの応用システムを紹介した。10MBPS 程度のネットワークの限界は明かに存

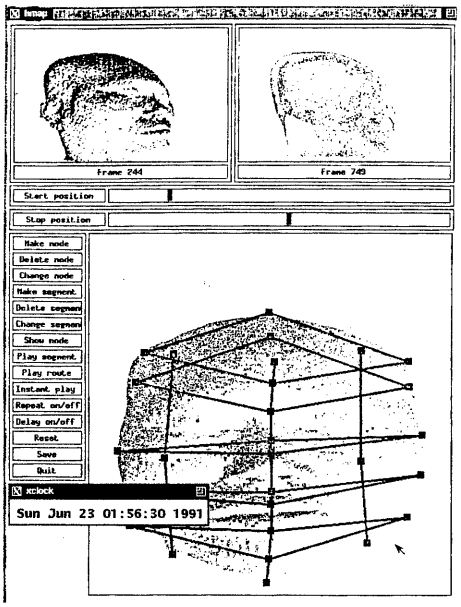


図 7: HyperMap の CG 画像再生への応用

在するが、現在の Unix ワークステーションの速度とのバランスを考慮すると決して絶望的に遅いものでないことが確認できた。ネットワークの通信容量もワークステーションの MIPS も必要がある限り伸び続けると考えるのが自然である。現在いくつかの大学で設置されているキャンパス LAN においても幹線は 100MBPS 以上の帯域を持っており、末端までその帯域を保証することも技術的に可能である。そのようなネットワークインフラと 100MIPS 級のワークステーションが普及した場合には今回の実験の 10 倍のレスポンスが得られることになり、画像メディアをとりまく計算機環境が一変すると考えるのが自然である。そういった環境でアプリケーションを開発するための手法、プログラミングパラダイムを考える必要があると思う。

# 参考文献

[1] 出口他, "コンピュータグラフィクスシステム LINKS-1 における画像生成の高速化手法", 情報処理学会論文誌 Vol.25,No.6,pp.944-952(1984)

[2] 山本, "計算量から見たレンダリングアルゴリズムの比較", 電子通信学会技術研究報告 IE90-98,pp. 17-24(1991)

[3] Whitted,T. "An Improved Illumination Model for Shaded Display", *Comm. ACM*, Vol.23, No.6,pp.343-349(1980)