

マルチコンピュータシステムMC-1における画像生成手法 Image Generation Method in Multicomputer System MC-1

日高 教行

平井 誠

中瀬 義盛

Noriyuki HIDAKA

Makoto HIRAI

Yoshimori NAKASE

浅原 重夫

鷺島 敬之

Shigeo ASAHARA

Takayuki SAGISHIMA

松下電器産業(株) 無線研究所

Matsushita Electric Industrial Co., Ltd. Wireless Research Laboratory

The image generation method in multicomputer system MC_1 is described. MC_1 has a root computer and many node computers and each node computer consists of a 16-bit MPU and an APU which can perform 24-bit floating point operations at the speed of 12 MFLOPS.

The algorithm based on Z_buffer hidden_surface algorithm and ray_tracing algorithm can generate efficiently real 3_dimensional images considered shadow, reflections and transparency. Some speed_up techniques are adopted, such as shadow preprocessing, dynamic computational load distribution by computational load prediction of each block (a division of screen area.) Furthermore, the algorithm can handle directly polygons, and quadratic curved surface.

1. まえがき

コンピュータの高速化に伴い、リアルな3次元画像を生成することが可能となり、CAD/CAM, ビジュアル・シミュレーション, アニメーション等の分野において、盛んに利用されている。

しかしながら、リアルな3次元画像をリアルタイムで生成するためには、現在のスーパーコンピュータの約1000倍の演算速度が必要と考えられ、これに対処する一つのアプローチとしてマルチコンピュータ方式がある。〔1〕〔2〕〔3〕

ところで、3次元画像生成手法としてよく知られているものに、次の3手法が挙げられる。

(1) Z-バッファ法〔4〕

この手法は、各物体のスクリーン投影時に、各ピクセルごとにz方向の深さをバッファし、隠面消去を行う。このアルゴリズム

は単純でありハードウェア化が容易である。特に多角形図形の投影処理においては、DDAにより高速化可能である。しかし、光源に対する物体の位置関係を利用していないため、陰影処理にはアルゴリズムの変更が必要である。

(2) スキャンライン法〔4〕

スクリーン座標系において、物体とy軸方向のスキャンライン平面の共通部分を求め、それらの位置関係に基づいて隠面消去を行なう。この手法は3次元問題を2次元問題に帰着し、しかも連続したスキャンラインを処理するため、物体のコヒーレンシー (coherency) を利用でき、多面体の処理においては、十分に高速化が計れる。しかし、陰影表示や、反射・屈折処理にはアルゴリズムの変更を要する。

(3) レイトレーシング (ray tracing) 法〔5〕

各ピクセルからレイ (ray) を出し、トレースすることにより、隠面消去が自然に行える。輝度計算に光学的モデルを用いることにより、陰影処理、反射・透過光の処理もでき、リアルな3次元画像生成に向いている。更に交差判定可能な表現形式の形状はすべて直接処理できるため、自由度が高い。欠点としては膨大な処理時間を要する。

上記の手法は一長一短を有するため、これらを有機的に組合せ、下記の特徴を有する画像生成アルゴリズムを採用し、マルチコンピュータシステムMC-1上にインプリメントしたので、このアルゴリズムについて報告する。

本アルゴリズムの特徴を以下に述べる。

1. 隠面消去が、Z-バッファ法により高速に行える。
2. 2次曲面を多面体近似により直接扱

える。

3. 光学的モデルに基づき、反射・透過処理まで施したリアルな3次元画像が生成できる。
4. 物体の位置関係に基づいた影リスト前処理により、視点変化に対しては、高速化が図れる。
5. 自由曲面は、B-スプライン(B-spline) 曲面に基づいたパッチにより、スムーズシェイディング (smooth shading) 可能である。
6. 各ブロック (スクリーンの矩形分割) に映り込む物体を保持するブロックリストをあらかじめ求めておき、画像生成時にブロック単位の動的負荷分散を行うことにより、各NCの負荷を均等にすることができる。各NCは、この

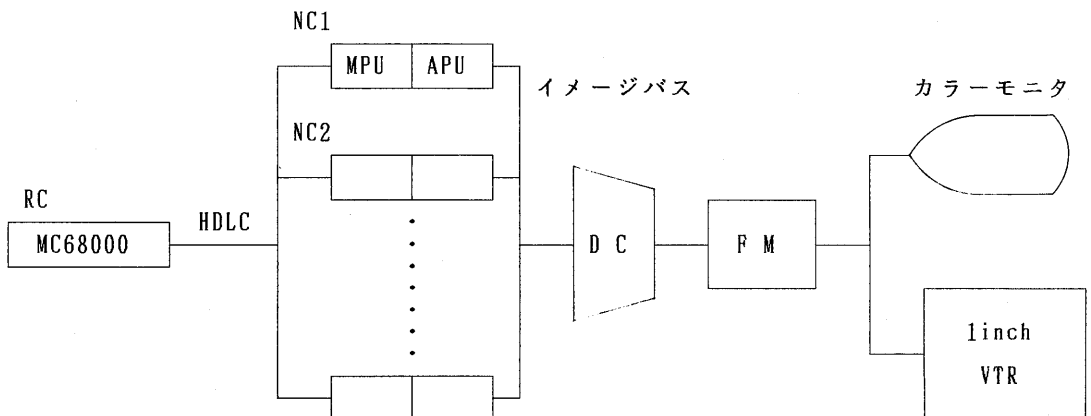


図1 MC-1 システム構成図

リストに基づいて本当に必要な物体のみを投影することにより、処理効率上がる。

2. 画像生成システム MC-1

図1はMC-1のシステム構成図である。MC-1は、ルートコンピュータ（以下RCと略す）とノードコンピュータ（以下NCと略す）がHDL C回線（4 Mbps）により図1の様に接続されたマルチコンピュータシステムであり、RCによる監視のもとで、各NCでは共通のプログラムが実行される。画像生成時、まずRCは、HDL C回線を用いてプログラム及びデータを一齐放送によりNCにダウンロードし、NC上のプログラムを起動する。その後RCは、ブロック（分割されたスクリーンの一部）単位の負荷を各NCに分散し、各NCの実行を監視する。各NCは、割り当てられたブロック内の各ピクセルの輝度を計算し、

専用の高速イメージバスに出力する。各NCからイメージバス上に出力された画像データは、データコレクタ（以下DCと略す）により集められフレームメモリ（以下FMと略す）に書き込まれる。

NCのAPUは、浮動小数点乗算器、加減算器、関数演算器（sin, cos, 逆数, 平方根）を持ち、それぞれ基本的に2ステージのパイプライン構成をとり、24ビットの浮動小数点演算を最大4 M回/秒実行できる。また、プログラムメモリ及びデータメモリはそれぞれ2バンクからなり、APUが演算を行っている間に、MPUは次の処理のためのプログラム及びデータを準備することができる。そのため、APUへのプログラム及びデータの転送時間を無視ことができ、APUとMPUは並行して動作する。

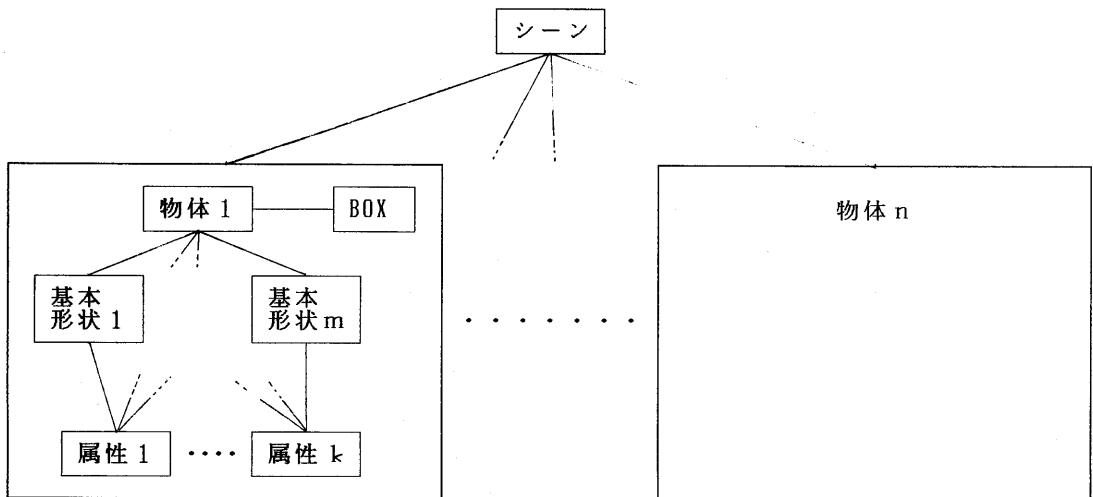


図2 物体データ構造

3. 物体データベース

画像生成プログラム実行時の物体データは、図2に示すように構成され、視点を原点とする座標系で記述されている。

各物体は、その物体を囲む $b o x$ と、その物体を構成する基本形状データと、その基本形状の表面属性を表わす属性データとからなる。

現在のところ、下記の基本形状が用意されている。

1. 三角板 (4角板)

- ・ 始点位置ベクトル及び辺ベクトル
- ・ 平面の法線ベクトル (物体の外側を向いている)

2. 疑似三角板 (疑似4角板)

- ・ 始点位置ベクトル及び辺ベクトル
- ・ 各頂点における法線ベクトル (物体の外側を向いている)

3. 楕円体

楕円体は一般に

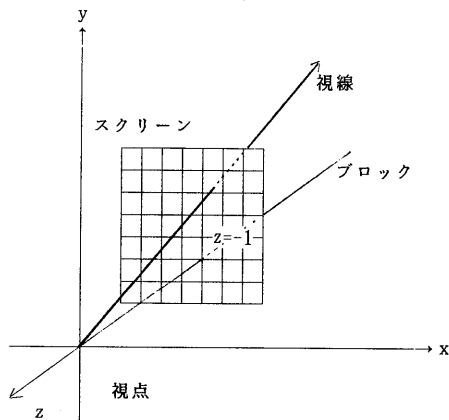


図4 視点座標系

$$|A(x-b)| \leq 1$$

なる式で記述されるが、その係数マトリクスと中心ベクトル b を保持している。

4. 一般2次曲面を表わす領域

$$a x + a y - a z + a x y + a x z + a y z + a x + a y + a z + a \leq 0$$

の係数 $a - a$ を保持している。

物体には2種類あり、ソリッド物体は、それを構成する基本形状の表わす領域の集合和として定義され、一方サーフィス物体は、これらの基本形状の表わす面を張りあわせた中空のサーフィスモデルである。多面体は、3角板、4角板の集合和として表現される。また、自由曲面で構成される物体は、B-spline曲面により補間され、疑似3角板及び疑似4角板の集合和として表現される。

一方、属性データは、各基本形状の光学属性、マッピングデータ、テクチャパターンを表わす。

4. 画像生成アルゴリズム

図3にMC-1での1シーンの画像生成手順を示す。既にNC上には、画像生成プログラムがダウンロードされているものとする。

① RCは実行型シーンデータを全てのNCにグローバル送信し、NCの画像生成プログラムを起動する。NCは物体及び光源を視点座標系に変換し(図4)、それぞれの物体を囲む $b o x$ を作成する。このとき、APUが一方のデータメモリバンクで座標変換している間、MPUが他方のデータメモリバンク上で、既に変換された物体データ、 $b o x$ データの回収及び次の物体

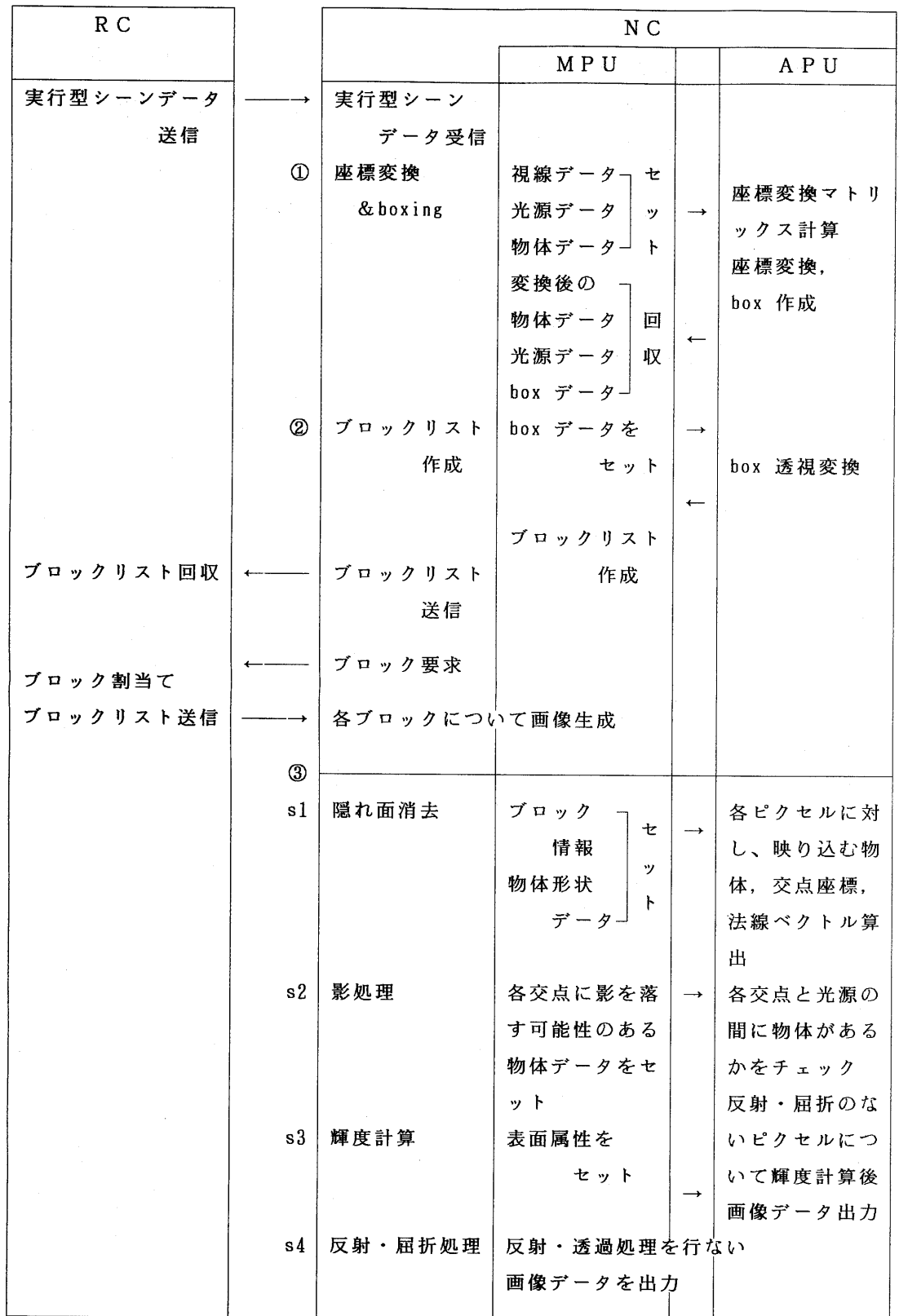
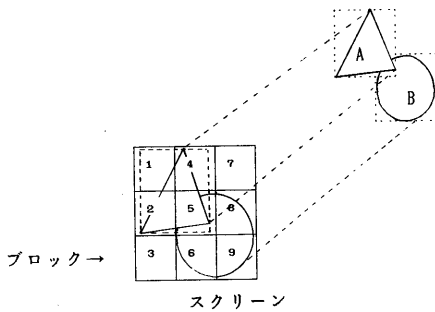


図3 MC-1における画像生成の流れ
(5)

データの書き込みを行うことができるため、APUへのデータの準備時間を無視することができ、効率良く座標変換を行うことができる。

② 各物体のboxをスクリーン上に投影し、それぞれのブロック（スクリーンを分割して得られる矩形，図2参照）に映り込む可能性のある物体の番号を、視点に近い順にソーティングしたリスト（以下、ブロックリストと呼ぶ）を作成する。図5に例を示す。

具体的には、RCが全スクリーンをブロックに分割し、NCにブロックを割当てる。各NCは割当てられたブロックに対してブロックリストを作成した後RCに送信し、RCは各NCから送られてくるブロックリストを回収する。



(a) スクリーンのブロック分割

ブロック番号

1	2	3	4	5	6	7	8	9
A	A		A	A	B		B	B
				B				

(b) ブロックリスト

図5 ブロックとブロックリスト

③ RCは、それぞれのNCに処理すべきブロックを動的に割付け、対応するブロックリストを各NCに送信する。以後RCは、ブロックリスト内の物体の個数及びその物体が反射・屈折属性をもつか否かに基づいてそのブロックの負荷予測を行い、負荷の大きいブロックから順に割付けていき、すべてのブロックの割付けが終了するまでこれを続ける。

各NCでは、割り付けられた各ブロックについて、隠面消去、影処理、輝度計算、反射・屈折処理を行い、画像データをイメージバスに出力する。

以下、各ステップの処理について詳述する。

s1：〔隠面処理〕

隠面処理では、MPUは、ブロックリスト中の物体データをAPUに転送する。APUは、その物体を構成する基本形状を順次ブロックに投影し、ブロック内の各ピクセルについて視線と視点に一番近い交点をもつ基本形状名とその交点のz値、その交点における法線ベクトルを記録する。

3角板、4角板は、各頂点がブロックに投影され、スキャンコンバージョン (scan conversion) により、その投影領域内のピクセルについてz値、法線ベクトルが求められる。x、y値が与えられるとz値、法線ベクトルは平面式から容易に求められる。疑似3角板、疑似4角板についても同様にしてz値が求められ、法線ベクトルは、各頂点の法線ベクトルから線形補間により決定される。

楕円体は、その楕円体に接するboxをまずスクリーンに投影し、その投影領域内

の各ピクセルに対し、視点とそのピクセルを結ぶ視線ベクトルと楕円体との交点を求め、その交点における法線を求めることにより処理される。その他の2次曲面についても同様である。

s 2 : [影処理]

M P U は、隠面処理で求めた各ピクセルに映り込んでいる物体のどれか1つに、影を落とす物体を順次 A P U に転送する。この処理は、あらかじめ作成された影リストに基づいて行われ、不要な物体の A P U 処理をなくし効率化している。影リストは、各物体について、光源からの直接光をささげる可能性のある物体名を保持している (図 6 参照)。

A P U はブロック内の各ピクセルについて、転送された物体が交点に影を落とす可能性がある場合には、交点と光源を結ぶ直線と転送された物体の基本形状との交差判定を行い、ある光源からの直接光をささぎ

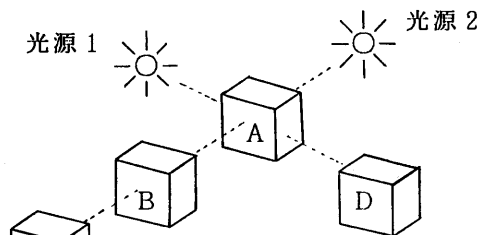
る場合には、その光源に対し影フラグをセットし、そのピクセルの輝度計算時には、直接光の計算を行わない (図 7 参照)。

s 3 : [輝度計算]

輝度計算では、M P U はブロックに映り込んだ物体の属性データを A P U に渡し、A P U は交点の座標、法線ベクトル、属性からピクセルの輝度を計算し、イメージバスに出力する。あるピクセルに映り込んだ物体が、反射・透過属性を持つ場合には、そのピクセルに対応する反射・透過フラグをセットし、画像データを出力せず後段で処理する。

輝度計算は、Whitted の光源モデルに基づいており、求める輝度は、R, G, B 3 成分をもつベクトル I で表現される。

$$I \leftarrow k_0 \begin{pmatrix} R \\ G \\ B \end{pmatrix} + k_d \begin{pmatrix} R \\ G \\ B \end{pmatrix} \times \sum_{l=1}^L I_l \cdot \cos \theta + k_m \begin{pmatrix} R \\ G \\ B \end{pmatrix} \times \sum_{l=1}^L I_l \cdot \cos \theta$$



物体 A の影リスト

光源 1	B C
光源 2	D

図 6 影リスト

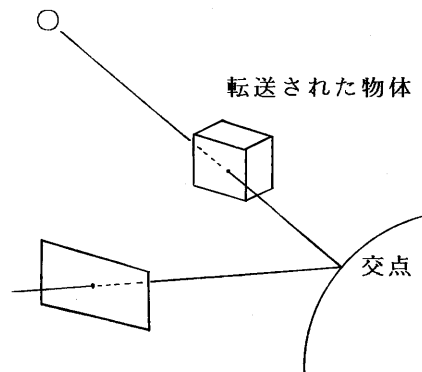


図 7 影処理

k_o : 周囲光係数

$\begin{pmatrix} R \\ G \\ B \end{pmatrix}$: 物体の色

k_d : 散乱反射係数

θ : 入射方向と法線のなす角度

L : 光源数

k_m : 金属反射係数

n : 金属反射指数

θ : 反射方向と視線のなす角度

ただし、"×"はベクトルの要素ごとの積を表わす。

第2項、第3項は、影処理において求めた影フラグに基づき、影フラグのたっていない光源についてのみ、直接光による輝度を加算する。

s 4 : [反射・透過処理]

輝度計算後、反射・透過フラグがセットされているピクセルについて、反射・透過処理を行う。ここでは、各交点を仮想的に視点とみなし、反射・透過方向の仮想的な視線に対し交差判定・影処理をくり返し輝度計算を行い画像データを出力する。

輝度計算式は次式で与えられる。

$$I \leftarrow k_o \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} + k_d \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \times \sum_{l=1}^L I_l \cdot \cos \theta \\ + k_m \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \times \sum_{l=1}^L I_l \cdot \cos \theta \\ + k_s \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \times S + k_t \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} \times T$$

k_s : 反射係数

k_t : 透過係数

S : 反射光による輝度

T : 透過光による輝度

7. 今後の課題

現在、MC-1は稼動したばかりであり、画像生成アルゴリズムの改良すべき点や追加すべき機能として、次の3項目が挙げられる。

1. 交差判定や、法線算出が容易な基本形状（たとえば回転体）の追加
2. 集合演算AND機能の追加
3. 画像生成プログラムの、各部の処理時間の評価実験を行い、プログラムの改良を行うこと、特に、APUの処理プログラムの並列度の改善が望まれる。

また、画像生成プログラムの評価実験を通じて、MC-1のハードウェアにも、改良を加えていく予定である。

最後に、本研究に際し有益なご助言をいただいた大阪大学工学部 電子工学科 大村助教授、白川助教授、京都大学工学部 情報工学科 富田助教授に感謝の意を表します。また、本研究の機会を与えていただいた当研究所 中島所長、三木部長に感謝します。

参考文献

1. 西村仁志他, "LINKS-I: コンピュータグラフィックスシステム", 情報処理学会マイクロコンピュータ研究会資料, 1982年11月.
2. Niimi H., et. al., "A Parallel Processor System for Three-Dimensional Color Graphics," ACM Computer Graphics, Vol. 18, No.3, pp. 67-76, (July, 1984).

3. Adam L., et. al., "Chap-A SIMD Graphics Processor," ACM Computer Graphics, Vol. 18, No.3, pp. 77-82.

(July, 1984).

4. Newman W. M. and Robert F. S., "Principles of Interactive Computer Graphics," 2nd Ed. McGRAW-HILL

5. Whitted, Turner, "An Improved Illumination Model for Shaded Display," CACM, Vol. 23, No.6, pp. 343-349, (June, 1980).



生成画像例