3DCG を描くための方法の開発

５年Ｂ組　川口恭平
指導教諭　末谷健志

１　要約

私は、3DCG（三次元コンピュータグラフィックス）を計算、描画するためのアルゴリズム、およびプログラムの研究を行い、既存の他アプリケーション、ライブラリ（DirectX,OpenGL 等）に依存せず 3DCG を描画することが出来るソフトの開発に成功した。

また、この成功により、3DCG 描画に関する基礎的な知識や技術を獲得でき、より高速、より高画質の 3DCG を描くための方法を考える足がかりを得ることが出来た。また、ヴァーチャルリアリティーの実現など、オリジナルの 3DCG 描画ソフトを応用したアプリケーションの開発が可能になった。

キーワード
3DCG、座標回転、透視図法、隠面処理、ワイヤーフレーム、レイトレーシング、Ｚバッファ

２　研究の背景

今日、パソコンの普及と高速化によって、3DCG を利用したコンテンツが増え、Shade など、市販のアプリケーションを使えば誰でも手軽に 3DCG を楽しむことが出来るようになった。3DCG の描画は、今後さらに拡大する可能性をもった技術である。このような中、私は、より高速でより高画質な 3DCG を描くための方法を研究したいと考えた。

しかし、3DCG を描くための方法やその処理は、アプリケーションが自動で行うため、ユーザーにとってはブラックボックスになっている。かといって、3DCG 描画の方法を調査するだけで、実際にプログラムなどに生かすことができなければ、むなしい調べ学習に終わるだけである。

そこで私は、3DCG 描画の勉強もかねて、他アプリケーションやライブラリに頼らず 3DCG を描画することができるソフトを開発することを試みた。

３　目的

他アプリケーションやライブラリに頼らず 3DCG を描画することができるエンジンを開発する。搭載する描画手法は、次の３つとする。

① ワイヤーフレーム法（図 1.1）
② レイトレーシング法（図 1.2）
③ Ｚバッファ法（図 1.3）

また、開発した描画エンジンの応用も検討する。

４　研究内容

（１） 仮説

仮説Ⅰ

ワイヤーフレーム法で3DCGを描画できる。

仮説Ⅱ

レイトレーシング法で3DCGを描画できる。

仮説Ⅲ

Ｚバッファ法で3DCGを描画できる。

上記仮説を検証するソフトを開発する環境は以下の通りである

OS：Windows XP　SP2

IDE: MicrosoftVisualBasic6.0
　　SP6 Enterprise Edition

（２） 研究方法

研究Ⅰ　ワイヤーフレーム法

与えられた頂点座標を平行移動、座標回転、透視図法処理を行ったの後、各頂点を線分で結び、画面上に表示する。

研究Ⅱ　レイトレーシング法

ベクトル方程式により直線（式1.1）や球（式1.2）を定義し、それらから交点の位置ベクトルまでの距離（相対）を導く式（式1.3）を得る。この式から、交点距離の比較を画素ごとに行い、隠面処理を実現する。

研究Ⅲ　光の効果について

① 環境光（図2.1）
② 局所拡散反射（図2.2）
③ 局所鏡面反射（図2.3）

の３種類の物理モデルをもとに、視点における光の見え方を計算し、3DCG として表現する。また、視点や光源の位置などから、陰影になる条件（図3.1～3.4）を導き、3DCGにおいてその結果を表示する。

研究Ⅳ　Ｚバッファ法

まず描画の対象となる図形をポリゴン（多角形）で分割し、図形をポリゴンの集合として考える。座標回転や透視図法処理にはポリゴンの頂点座標のみを考え、平面に投影後、多角形を塗りつぶす。塗りつぶす際に、Ｚバッファを用いて奥行きに関して比較を行い、隠面処理をおこなう。また、研究Ⅲにおいて実現した光の効果をＺバッファ法においても実現し、画面上に表示する。

＜隠面処理とは＞

人が物体を見る際、奥にある物体は手前にある物体によって隠される。このことは現実ではあたりまえであるが3DCGとして計算する場合、奥にある物体の手前にある物体によって隠された面を描かないようにする処理が必要である。この処理のことを隠面処理といい3DCG描画において重要な処理のひとつである。

＜Ｚバッファとは＞

図形を平面に投影する際、その画像情報のメモリとは別に用意する、奥行きに関する一時メモリのこと。

（３） 研究の結果

研究Ⅰ　ワイヤーフレーム法によって次の画像（図 4.1，図 4.2）を得た。

研究Ⅱ　レイトレーシング法によって次の画像（図 5.1）を得た。

研究Ⅲ　反射光における結果として次の画像（①図 6.1、②図 6.2、③図 6.3、①+②+③図 6.4）を得た。また、陰影処理の結果として次の画像を得た（6.5）

研究Ⅳ　Ｚバッファ法によって次の画像（図 7.1）を得た。また、光の効果を考慮にいれた結果として次の画像(図 7.2)を得た。

以上の結果から、仮説Ⅰ，Ⅱ，Ⅲを実証することが出来た。

５　考察

―研究Ⅰ―

実現がたやすく、処理が単純なので高速で、簡易的な 3DCG として様々なものに応用できそうである。しかし表現力に乏しく、位置や形状の正確な把握が難しく、フォトリアリスティックな利用には向いていない。出来るだけ高速な処理を必要とするシミュレーションや設計などでの利用が考えられる。

―研究Ⅱ―

結果の画像から、隠面処理が出来ているのがわかる。すべての画素ごとに距離を計算しているので、視点を近づけても得られる画像は荒くならないが、計算量が多く、描画に非常に時間がかかることが問題である。

―研究Ⅲ―

反射光における結果の画像から、光の効果によって物体表面の色が変化しているのがわかる。しかし利用した物理モデルは、厳密な解ではなく、あくまで近似であるため、多少の不自然さが画像にも見える。また、現実にある素材の見え方の再現をおこなうための各設定値（拡散反射係数など）を決定するのが難しい点などから、今回実現できた光の反射光の効果はあくまで擬似的なものであるといえる。このことから、より現実に近い光の反射光の表現、また、映り込みや、屈折現象などの高度な光の効果の表現が課題として考えられる。

陰影処理の結果の画像からは、陰影処理によって物体に陰影がついたのが確認できる。現実の世界では影は境界がぼやけているが、それは光源が有限の大きさを持っているからであり、光源を点光源として計算したので今回の画像は影の境界がシャープになったと考えられる。このことから、次の研究では有限の大きさを持った光源についての影の出来方について研究が考えられる。

―研究Ⅳ―

レイトレーシング法に比べると、処理そのものは複雑だが、画素毎ではなく図形毎に処理を行うので、比較的高速に処理が出来た。画質については、結果の画像（図 6.2）を見てもポリゴンに分割したことによる角ばりはあるものの、あまり差は無いようにみえるが、視点を図形に近づけた場合、ポリゴンの分割数が少ないと、角ばりが大きく出てしまう。また、透視図法処理は頂点座標のみに行われるので、面については透

視図法処理による効果が考慮されず、ポリゴンの大きさが大きくなると、それが考慮されないことによるゆがみが増大するので、必要に応じてポリゴンによる図形の分割数を多くする必要があると考えられる。

光の効果を考慮して描画した結果をみると、明らかに表面がゴツゴツしているのが分かる。これは、描画高速化のため、図形を多角形で分割したためであると考えられる。また、陰（自身による光の遮りによって生じる暗闇）の効果は出ているが、影（他の物体による光の遮りによって生じる暗闇）の効果については反映されないことがわかる。以上のことから、レイトレーシング法と比較したとき、隠面処理についてはあまり差は出ないが、その後の光の効果を表現する処理について、大きく差が出ることが分かった。

―全体を通して―

今回実現できた3DCG描画方法の特徴を次の表(表1)のようにまとめることが出来る。これらの3DCG描画方法は表から読み取れる通り、他のものに秀でている部分と劣っている部分をもっている。たとえばレイトレーシングとＺバッファの性能を比べると、レイトレーシングは光の効果の再現や画質でＺバッファに勝るが、描画に非常に時間がかかるという欠点があり、Ｚバッファは高速に描画できるが、画質があまりよくないという欠点がある。もちろんＺバッファも画質を高めることができ、Ｚバッファでは曲面をポリゴンで分割するため、描画した画像の曲面もカクカクしているが、分割せずなめらかな曲線が描画可能なことが確かめて分かっている。しかし、処理が複雑になるばかりか、その分処理が遅くなってしまい、画質だけならそれでもレイトレーシングが圧倒的に勝っているので、Ｚバッファのメリットがなくなってしまう。このことから分かるとおり、それぞれの3DCG描画方法は向き不向きがあり、用途に応じて、適切な3DCG描画方法を選ぶべきだということが分かる。

６　まとめと今後の課題

今回の研究において3種類の3DCG描画するためのアルゴリズム、およびプログラムの研究を行い、既存の他アプリケーション、ライブラリ（DirectX,OpenGL 等）に依存しない3DCG描画ソフトの開発に成功した。開発に成功し3DCG描画に関する基礎的な知識や技術を獲得できたことにより、より高速、より高画質の3DCGを描くための方法を考える足がかりを得ることが出来た。またオリジナルの3DCG描画ソフトを応用したアプリケーションの開発が可能になった。

今回実現できた3種類の3DCG描画アルゴリズムはどれも有名なものばかりで、無論自分自身で開発したものではない。これからの課題としては、これらの3DCG描画アルゴリズムの改良や新たな3DCG描画アルゴリズムの開発をしていくことが挙げられる。具体的な例としてはＺバッファとレイトレーシングを組み合わせ、双方の利点を得られるような新たな3DCG描画アルゴリズムを現在開発中である。また、今回実現できたソフトの応用として、モーションキャプチャーと組み合わせることによるヴァーチャルリアリティーや、多次元シミュレーションの結果表示などを模索している。

７　参考文献・サイト

http://www.iamas.ac.jp/~tacwon/render.html#history

８　謝辞

今回の研究に協力してくださった末谷先生や河合先生、佐久間先生にこころよりお礼申し上げます。

□図、式、表□

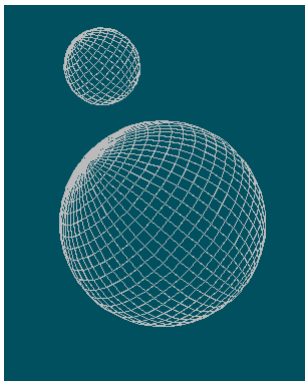


図 1.1

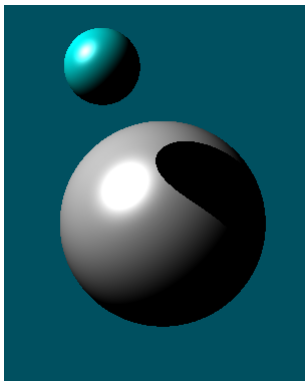


図 1.2

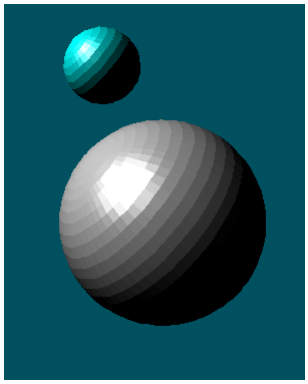


図 1.3

$\vec{p}$:視線上の点の位置ベクトル

$\vec{e}$:視点からスクリーン上の一点までの方向ベクトル

$\vec{c}$:球の中心の位置ベクトル

$\vec{E}_0$:視点の位置ベクトル

$$\vec{p} = \vec{e}t + \vec{E}_0$$

式 1.1

$$(\vec{p} - \vec{c}) \cdot (\vec{p} - \vec{c}) = r^2$$

式 1.2

$$t = \frac{-\beta \pm \sqrt{\beta^2 - \alpha\gamma}}{\alpha}$$

ただし $\left\{ \begin{array}{c} \alpha = |\vec{e}|^2 \\ \beta = \vec{e} \cdot (\vec{E}_0 - \vec{c}) \\ \gamma = |\vec{E}_0 - \vec{c}|^2 \end{array} \right\}$

式 1.3

環境光反射の数理モデル

L_i :入射する光の強さ
L_o :観測する光の強さ

$$L_o = L_i$$

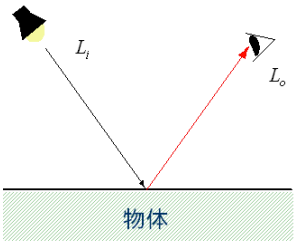


図 2. 1

拡散反射の数理モデル

$\vec{N}$:面の法線ベクトル
$\vec{I}$:光線ベクトル
θ : $\vec{I}$ と $\vec{N}$ がなす角
L_i :入射する光の強さ
L_o :観測する光の強さ

Lambert余弦則

$$L_o = (\vec{I} \cdot \vec{N}) L_i = L_i \cos\theta$$

（ただし $\vec{I}$ と $\vec{N}$ は共に単位ベクトル）

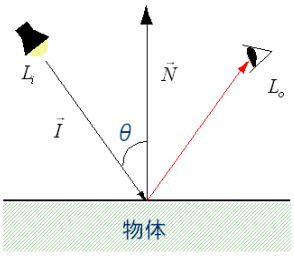


図 2. 2

鏡面反射の数理モデル

$\vec{N}$:面の法線ベクトル
$\vec{I}$:光線ベクトル
$\vec{R}$:反射ベクトル
$\vec{V}$:視線ベクトルの逆ベクトル
L_i :入射する光の強さ
L_o :観測する光の強さ

$$\vec{R} = 2(\vec{I} \cdot \vec{N})\vec{N} - \vec{I}$$

Phong鏡面反射

$$L_i = (\vec{R} \cdot \vec{V})^n L_o = L_o \cos^n \phi$$

（nはハイライトの特性を決める定数）

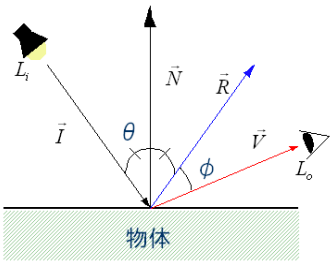
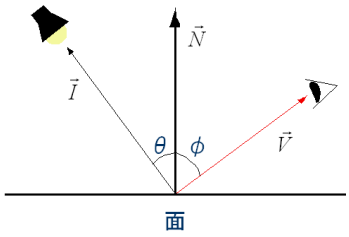


図 2. 3

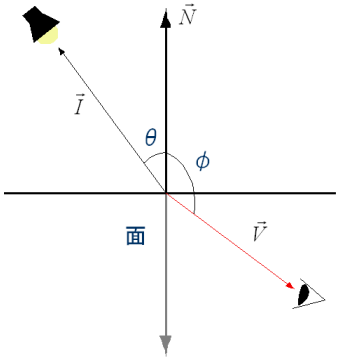
陰にならないパターン



$$\pm\cos\theta \geq 0$$
$$\pm\cos\varphi \geq 0$$
$$\therefore \cos\theta\cos\varphi \geq 0$$
$$\Leftrightarrow (\vec{N} \cdot \vec{I})(\vec{N} \cdot \vec{V}) \geq 0$$

図 3. 2

陰になるパターン



$$\pm\cos\theta \geq 0$$
$$\mp\cos\varphi \geq 0$$
$$\therefore \cos\theta\cos\varphi \leq 0$$
$$\Leftrightarrow (\vec{N} \cdot \vec{I})(\vec{N} \cdot \vec{V}) \leq 0$$

図 3. 2

影になるパターン

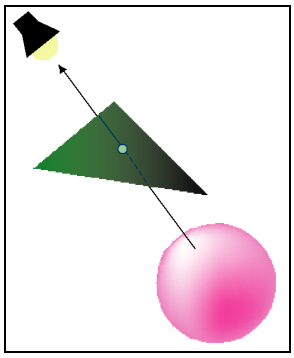


図 3.3

影にならないパターン

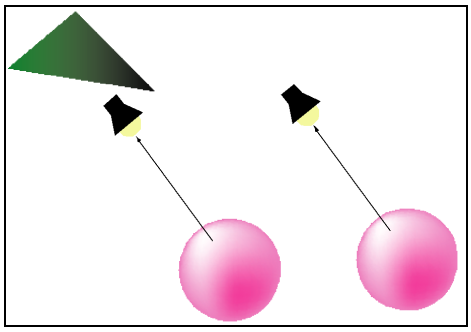


図 3.4

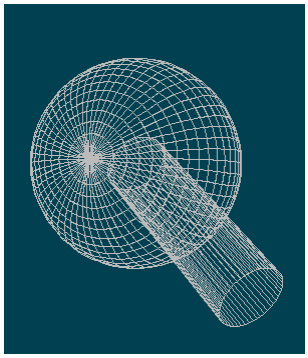


図 4.1

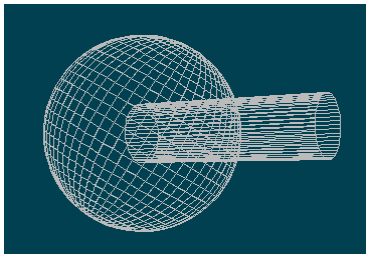


図 4.2

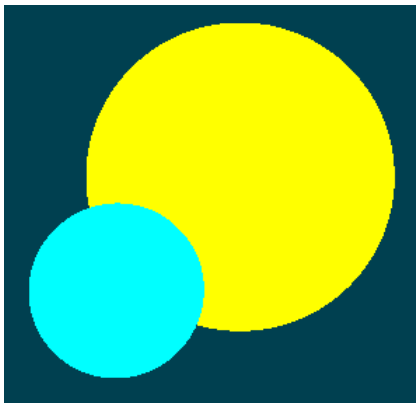


図 5.1

①周囲光

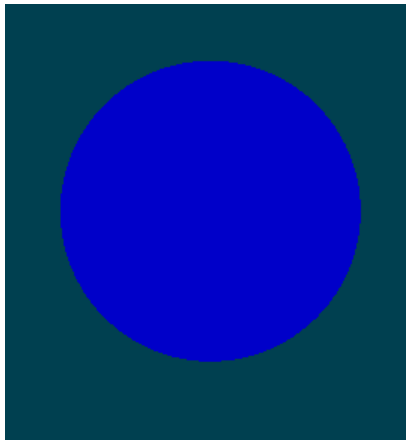


図 6.1

②局所拡散反射光

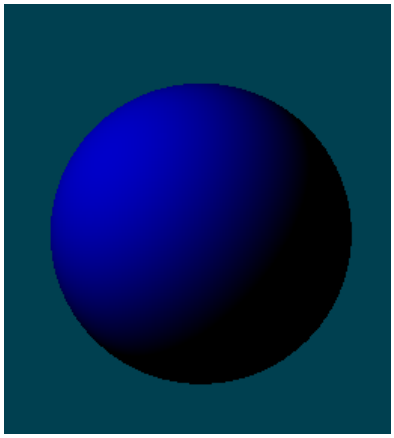


図 6.2

③局所鏡面反射光

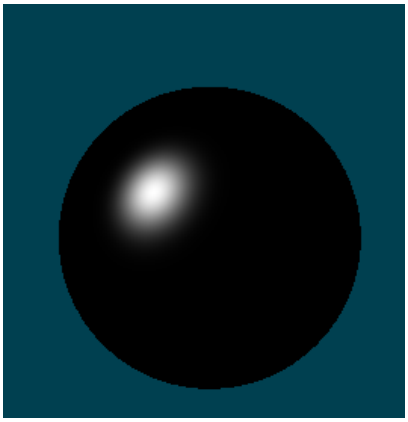


図 6.3

①②③統合

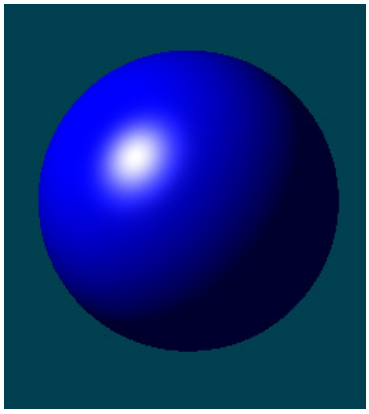


図 6.4

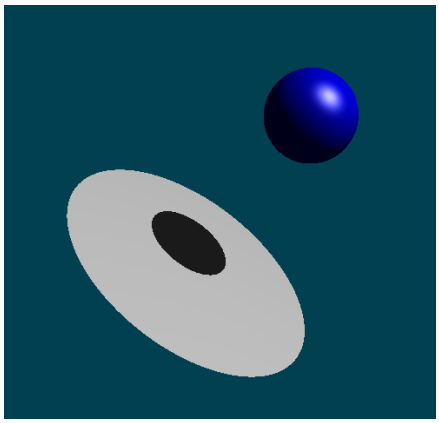


図 6.5

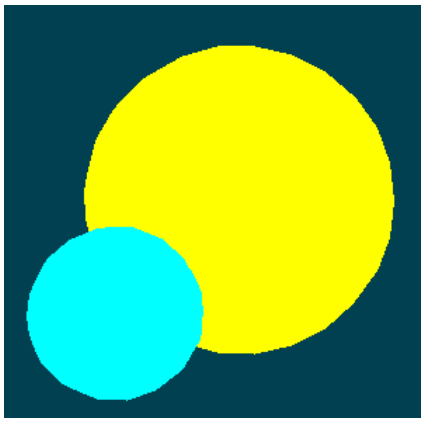


図 7.1

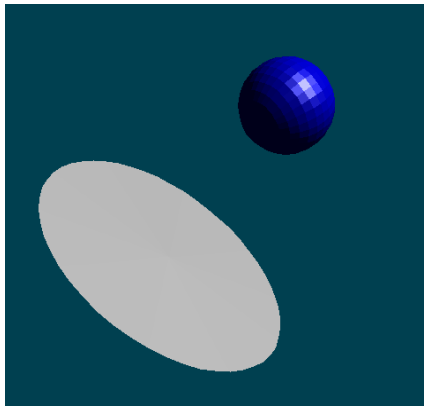


図 7.2

項目	ワイヤーフレーム	レイトレーシング	Zバッファ
処理速度	◎	×	○
面の表現	×	○	△
光の効果	×	○	△
陰影の効果	×	○	△
モデリングのしやすさ	○	△	○

表 1