修士学位論文

表面構造を考慮した
複眼のリアルタイムレンダリング

平成 26 年度

東京大学大学院 学際情報学府

先端表現情報学コース

136313

佐川 和輝

指導教員　河口 洋一郎 教授

目次

図目次

表目次

第 1 章

序論

1.1　研究背景

近年では、コンピュータグラフィックスの表現において、実写並みのリアリズムが要求されるようになってきた。実物と見間違うほどの CG も珍しくなくなり、聴衆は実物と CG との差異に敏感になってきている。

1.2　本研究の目的

これまでに紹介したように、技術の進歩にともなって、コンピュータグラフィックスに対する要求は増してきつつある。さらに、近年では実時間での計算

1.3　本論文の構成

＊＊＊要チェックポイント＊＊＊

本論文の構成について述べる。次の第 2 章では、本研究と関連のある技術手法や生物分野で複眼について調査等を行った研究を紹介する。第**??**章では、周辺技術として本研究の基礎となるシェーダアルゴリズムについて解説を行う。続いて第 4 章では、過去の研究に基づいて本研究で実際に行った予備実験について説明する。実験結果を踏まえて第 6 章では、新しく提案するシミュレーション手法を述べる。

第 2 章

関連研究

りれーてっどわーく。てすとてすとてすとテスト

2.1　微細構造を扱った研究

表面微細構造をもつ物質をレンダリングする研究が過去にいくつか行われている。

2.2　本研究の位置づけ

本研究の位置づけについて述べる。

第 3 章

周辺知識

本章では、……と関連の深いなんとかかんとか。

第 4 章

予備実験

4.1　実験の目的

実際の複眼の構造や形状は複雑であり、コンピュータグラフィックスにおいて実物の形状を作成したり、実物の構造計算に用いることは難しい。そこで、複眼の性質を表現するために代替となる形状や近似モデルが必要となる。本研究では、永田 [] が行った偽瞳孔の再現実験をもとにビー玉とパンチングメタルを用いて予備実験を行った。

本実験における第一の目的は、 偽瞳孔の発生原理を理解し、物理現象に落としこむことである。複眼を扱った関連研究のうち、偽瞳孔の性質や現象について生物学上の考察や解説を行うものは存在するものの、幾何学的なしくみについて言及したものは少ない。そのため、実物を観察することによって物理現象としてのしくみを明らかにし、開発の足がかりとする必要があった。第二の目的は、実装を行う前に近似手法による偽瞳孔の再現度を確認することである。永田の実験は複眼の形状を大きく変え、平板と球体として近似している。そのため、コンピュータグラフィックスとして表現するにあたり、適切な手法であることを確認することが望ましい。そして第三の目的は、レンズや色素細胞に相当するパンチングメタルの穴などの大きさや周期の違いが模様に与える変化を観察することである。実際の昆虫などの複眼は各個眼の大きさが決まっており、全体の形状に対して自由に個眼の大きさを変化させることができない。ゆえに、大きさや周期といったパラメータの変化に対して偽瞳孔の模様が変化する様子を確認するために模型を利用する。

4.2　実験方法

実験に用いた道具は以下のとおりである (Fig.**??**)。

- ビー玉（透明なもの）
- パンチングメタル（穴の大きさと周期の違うもの 2 種類）
- 黒色の画用紙
- 木の棒

まず、画用紙とパンチングメタルをセロハンテープなどで固定する。これは、偽瞳孔の模様をはっきりとさせるためにパンチングメタルの穴の部分を黒く見せるためである。次に、木の棒で枠を作りパンチングメタルの上に乗せる。続いて、木の枠内にできるだけ隙間が開かないようにビー玉を敷き詰める。このモデルでは、パンチングメタルの穴が複眼の色素細胞に相当し、ビー玉が個眼のレンズに相当する。実際の複眼では個眼のそれぞれにおいて色素細胞とレンズは対応しており同じ周期で配置されているが、本実験では厳密にパンチングメタルの穴の位置とビー玉の配置を一致させてはいないが、密集した球体レンズによる光の屈折がどのような虚像を生み出すのかを確認するためには十分である。

4.3　結果と議論

第 5 章

本手法の基礎技術

5.1　グラフィックスパイプライン

5.2　オブジェクトファイルフォーマット

モデリングソフト等により前もってオブジェクトファイルを作成し、アプリケーションで読み込みを行う。さらに、物体形状および法線やテクスチャ座標などの情報を元にアプリケーション内でシェーダへの転送データを生成する。本手法で必要になるオブジェクトファイルのデータは以下の要素である。

- 頂点座標値
- 頂点法線ベクトル
- テクスチャ座標値
- 面情報

頂点座標値および頂点法線ベクトルはそれぞれ 3 次元の浮動小数点型、テクスチャ座標値は 2 次元の浮動小数点型の値となっている。面情報は、頂点座標値、頂点法線ベクトル、テクスチャ座標値のインデックス番号の組み合わせとなっており、面が表す多角形の頂点の数だけこれらの情報が与えられている。本手法では三角形ポリゴンのみを対象としているため、この組み合わせが 3 つずつ続く。

5.3　GPU 処理

5.3.1　シェーダ

画像処理用演算プロセッサ（GPU:Graphics Processing Unit）で処理を行うのはシェーダとよばれるプログラムであり、このシェーダよって演算処理が実行される。シェーダ（shader）ではライティング（光源計算）、シェーディング（陰影処理）およびレンダリング（画像ピクセル化）を行う。本手法では、リアルタイム処理を行うため、プログラマブルパイプライン

第 6 章

提案手法

1.2 節で述べたように、本研究の目的は複眼表面に観測される光学的現象をリアルタイムレンダリングによって表現することである。本章では、レンダリング時の計算アルゴリズムについて詳細に解説していく。まず、6.1 節で本手法で用いた複眼の近似モデルを提案する。次に 6.3 節では、アプリケーションから画像処理用演算プロセッサ（GPU）へと転送するデータと、その生成時に用いた計算アルゴリズムを説明する。そして 5.3 節では、画像処理用演算プロセッサにおいて描画色を求める手法を説明する。

6.1　フレームワーク

6.1.1　複眼モデル

第 4 章で取り上げた実験をもとに、シェーダアルゴリズムに図のようなモデルを採択した(Fig.6.1)。ポリゴン直下に屈折レンズの役割を果たす球体を配置し、表面を埋め尽くすように多数配置している。さらに、テクスチャ平面と称してポリゴンと平行な位置にテクスチャ情報を取得するための平面を配置している。テクスチャ平面は、**??**節で説明した色素細胞の役割を担っている。レンズによって屈折された光はテクスチャ平面へ到達し、テクスチャ平面上の情報から色への影響を決定する。

続いて、モデルの利点を説明する。まず、このモデルは複雑な複眼の構造を球や平面などの単純な幾何立体の集合として扱うことができる。そのため、光の屈折等を計算する際に、きわめて軽量な計算量で済むという利点がある。次に、第 4 章の実験は永田 [] が示しているように偽瞳孔現象の特徴を十分に再現している。ゆえに、近似モデルとして十分に目的を果たすことが期待できる。

Fig. 6.1　シェーディングモデル

- ポリゴンの直下にレンズとして球体を配置。
- 距離を開けてポリゴンと平行にテクスチャ平面を配置。

6.1.2　アルゴリズムの全体像

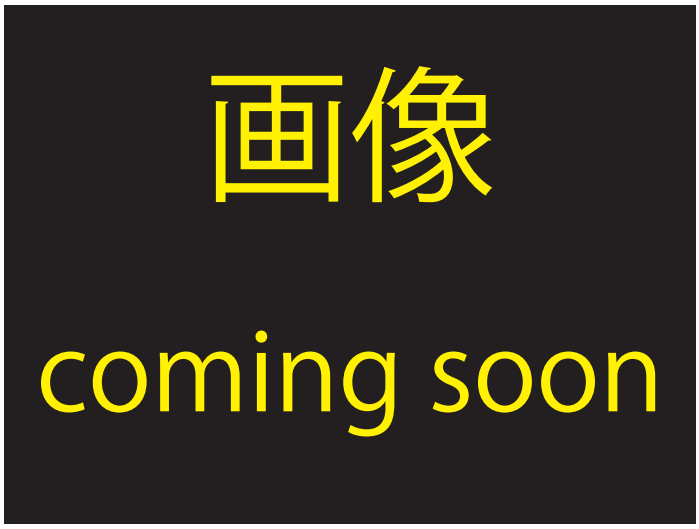

Fig. 6.2　アルゴリズムフレームワーク

本手法のアルゴリズムの全体像を説明する。本手法の構成は大きく分けると、最初に複眼表面を適用する形状データの作成、次にアプリケーションにおいて計算に用いるデータの処理。そして最後に偽瞳孔による光の減衰量計算および光源、陰影処理等がなされる (Fig.6.2)。アプ

リケーションでは通常の手法で作成されたオブジェクトファイル情報を加工し、計算に必要な変数を GPU（シェーダ）へ送る。シェーダでは屈折などを考慮した物理計算を行い、最終的に画面に描画する色情報を計算する。

6.2　テクスチャと球の配列

テクスチャおよび球は複眼における個眼をモデル化したものである。そのため、実際の個眼と同様にひとつのテクスチャ単位とひとつの球の位置を合わせて配置する必要がある (Fig.6.3)。

Fig. 6.3　テクスチャと球の位置関係

今回用いた手法では、格子状に球を配置させてテクスチャとの位置合わせをおこなう (Fig.6.4)。具体的には、テクスチャとして用いている正方形画像の各頂点と格子の各点を対応させ、テクスチャ座標軸方向と格子の平行線を一致させる。正規化されたテクスチャをタイル状に繰り返し適用し、格子の中心と面の法線方向から見た球の中心座標を一致させる。すなわち、法線方向の奥行きは一致しておらず間隙がある。球の半径を r とすると、ポリゴンとテクスチャ平面との距離は $2r$ となる (Fig.**??**)。

正確には、色素細胞として用いるテクスチャ座標値と球の配列に利用するテクスチャ座標値は独立して扱うことができるため、必ずしも同一のものを利用する必要はない。ここでは、両者を同一化しても計算上問題が無いためそのまま利用する。

Fig. 6.4　テクスチャおよび球の格子状配置

6.3　計算用データ

読み込んだオブジェクトファイルデータから計算用のベクトルデータを生成する。この処理はアプリケーション内で一度だけ行われ、プログラマブルシェーダへ転送されたのち保持される。頂点情報としてにシェーダへ転送されるデータは、頂点座標値、頂点法線ベクトル、テクスチャ座標値、そして後述する接ベクトル情報である。これらのうち頂点座標値および頂点法線ベクトルは更新されず、オブジェクトファイルから読み取った値を面情報にしたがって順次バーテックスバッファオブジェクトの形で配列情報として転送される。テクスチャ座標値はユーザ指定の浮動小数点型の値であるテクスチャ解像度 R_t を以下のように乗算し転送される。

$$\boldsymbol{T}_{vert} = R_t \boldsymbol{T}_{obj} \tag{6.1}$$

ここで、$\boldsymbol{T}_{obj}$ は作成した形状データから読み込んだテクスチャ座標値、$\boldsymbol{T}_{vert}$ は頂点シェーダ（5.3.1 節）へ転送されるテクスチャ座標値である。

本手法ではテクスチャ座標値をもとに屈折レンズ（6.1.1 節）の配置を決定しているため、R_t を変更することで複眼表面のレンズの配置すなわち表面構造の細かさを任意に変更することができる。これらの頂点情報の他にシェーダへ与えられる定数などの情報はプログラマブルシェーダ内の変数として適宜転送される。

6.3.1 テクスチャ座標軸方向 3 次元単位ベクトル

通常、2 次元空間上のデータであるテクスチャは、テクスチャ座標値と空間上の点との対応づけにより 3 次元空間上に描写される。すなわち、3 次元空間における面は 2 次元の座標空間として考えることができる (Fig.**??**)。そこで、3 次元空間内の情報から 2 次元情報であるテクスチャ座標値を算出するためには、異なる次元同士を橋渡しする変数が必要になる。

本手法では、オブジェクト上の各位置におけるテクスチャ座標系の単位ベクトルを 3 次元ベクトルとして表すことによって、異なる次元の情報を結びつけている。本項では、2 次元ベクトルであるテクスチャ座標空間の単位ベクトルを、3 次元空間上の 3 次元ベクトルとして表す方法について述べる。また、3 次元空間上のテクスチャ座標軸方向単位ベクトルを用いると、ポリゴン上の任意の点においてテクスチャ座標値を逆算できるようになる。

ポリゴン上のある位置におけるテクスチャ座標空間の軸方向の単位ベクトルを 3 次元ベクトル $\boldsymbol{U}_p, \boldsymbol{V}_p$ として表すと、ポリゴン上の任意点 $\boldsymbol{P}$ は $\boldsymbol{U}_p, \boldsymbol{V}_p$ を利用して以下のように表現することができる。

$$\boldsymbol{P} = \boldsymbol{P}_c + a\boldsymbol{U}_p + b\boldsymbol{V}_p \tag{6.2}$$

ここで、$\boldsymbol{P}_c$ は既知の点 $\boldsymbol{P}_e$ およびそのテクスチャ座標値 (a_e, b_e) によって

$$\boldsymbol{P}_c = \boldsymbol{P}_e - a_e\boldsymbol{U}_p - b_e\boldsymbol{V}_p \tag{6.3}$$

式 (6.3) のように表される。また、式 (6.2) の a および b は点 $\boldsymbol{P}$ におけるテクスチャ座標値 (a, b) を表している。すなわち、3 次元空間上のテクスチャ座標軸方向単位ベクトル $\boldsymbol{U}_p, \boldsymbol{V}_p$ が既知であれば、式 (6.2) の係数を利用してポリゴン上の任意の点におけるそのテクスチャ座標値 (a, b) を逆算することが可能になる。本手法では、シェーダでの処理に 3 次元ベクトル $\boldsymbol{P}_c$, $\boldsymbol{U}_p$ および $\boldsymbol{V}_p$ が必要となるため、これらの値を作成しシェーダへ転送する必要がある。

3 次元単位ベクトル計算手順

テクスチャ座標軸方向 3 次元単位ベクトル $\boldsymbol{U}_p$ および $\boldsymbol{V}_p$ は以下の手順で求める。$\boldsymbol{U}_p$ および $\boldsymbol{V}_p$ はポリゴン毎に変化するベクトル変数であり、ポリゴンを構成する各頂点の頂点情報としてバッファに格納される。まず、本手法で対象としている三角形ポリゴンの各頂点の頂点座標値を $\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2$ とし、テクスチャ座標値を $\boldsymbol{T}_0, \boldsymbol{T}_1, \boldsymbol{T}_2$ とする。ここでは、頂点座標値が 3 次元ベクトルであるのに対してテクスチャ座標値が 2 次元ベクトルであることに留意し、ポリゴ

ン上で両者の対応関係を明確にしていく。

三角形ポリゴンは 3 次元空間上における平面を一意に表すことができるため、この平面に対応するベクトルを $\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2$ および $\boldsymbol{T}_0, \boldsymbol{T}_1, \boldsymbol{T}_2$ から求めていく。

Fig. 6.5　ポリゴン上における頂点座標値およびテクスチャ座標値の関係

頂点座標値から相対ベクトル $\boldsymbol{P}_{10}$,$\boldsymbol{P}_{20}$ を以下のように定義する。

$$\boldsymbol{P}_{10} = \boldsymbol{P}_1 - \boldsymbol{P}_0 \tag{6.4}$$

$$\boldsymbol{P}_{20} = \boldsymbol{P}_2 - \boldsymbol{P}_0 \tag{6.5}$$

同様に、テクスチャ座標値から相対ベクトル $\boldsymbol{T}_{10}$,$\boldsymbol{T}_{20}$ を以下のように定義する。

$$\boldsymbol{T}_{10} = \boldsymbol{T}_1 - \boldsymbol{T}_0 \tag{6.6}$$

$$\boldsymbol{T}_{20} = \boldsymbol{T}_2 - \boldsymbol{T}_0 \tag{6.7}$$

これらの相対ベクトルはポリゴンのエッジに相当し、それぞれ平面上の変位を表すベクトルとなっている (Fig.6.5)。

さて、テクスチャ座標軸方向 3 次元単位ベクトル $\boldsymbol{U}_p$ および $\boldsymbol{V}_p$ は i, j, k, l を適当な係数として以下のように表すことができる。

$$\boldsymbol{U}_p = i\boldsymbol{P}_{10} + j\boldsymbol{P}_{20} \tag{6.8}$$

$$\boldsymbol{V}_p = k\boldsymbol{P}_{10} + l\boldsymbol{P}_{20} \tag{6.9}$$

そして、係数 i, j, k, l は $\boldsymbol{T}_{10}$ および $\boldsymbol{T}_{20}$ から導くことができる。

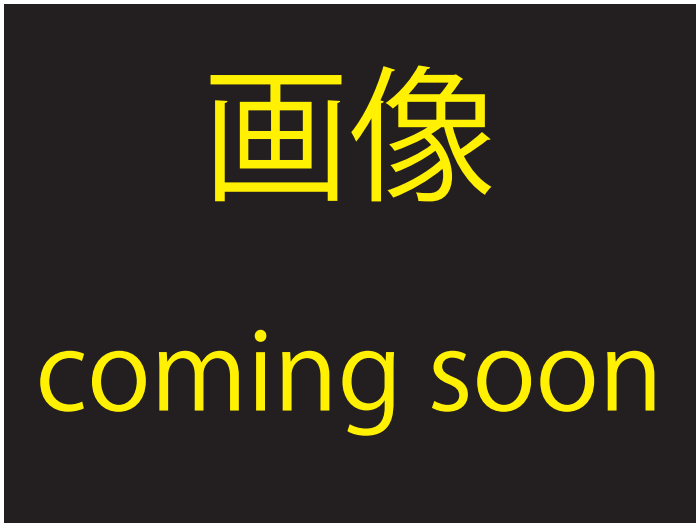

Fig. 6.6　平面上の変位ベクトル

続いて、ポリゴン平面上の任意の位置にある点 A および B を考える (Fig.6.6)。点 A の頂点座標値を $\boldsymbol{P}_A$ そして点 B の頂点座標値を $\boldsymbol{P}_B$ とすると、点 A および点 B の 3 次元空間上における位置の変位は、c, d を適当な係数として以下のように表すことができる。

$$\begin{aligned} \boldsymbol{P}_{AB} &= \boldsymbol{P}_A - \boldsymbol{P}_B \\ &= c\boldsymbol{P}_{10} + d\boldsymbol{P}_{20} \end{aligned} \tag{6.10}$$

さらに、$\boldsymbol{P}_{10}$ と $\boldsymbol{T}_{10}$ および、$\boldsymbol{P}_{20}$ と $\boldsymbol{T}_{20}$ がポリゴン上で対応関係にあることから、点 A のテクスチャ座標値を $\boldsymbol{T}_A$ そして点 B のテクスチャ座標値を $\boldsymbol{T}_B$ とすると以下の式が成り立つ。

$$\begin{aligned} \boldsymbol{T}_{AB} &= \boldsymbol{T}_A - \boldsymbol{T}_B \\ &= c\boldsymbol{T}_{10} + d\boldsymbol{T}_{20} \end{aligned} \tag{6.11}$$

ここで、i, j, k, l を用いて

$$i\boldsymbol{T}_{10} + j\boldsymbol{T}_{20} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{6.12}$$

$$k\boldsymbol{T}_{10} + l\boldsymbol{T}_{20} = \begin{pmatrix}0\\1\end{pmatrix} \tag{6.13}$$

とし、テクスチャ座標空間を UV 座標で表すと、2 次元空間内において式 (6.12) は U 軸単位ベクトル、式 (6.13) は V 軸単位ベクトルを表すことになる。さらに、式 (6.10) および式 (6.11) の対応関係から式 (6.8) および式 (6.9) を導くことができる。

2 次正方行列 $\boldsymbol{A} = (\boldsymbol{T}_{10}, \boldsymbol{T}_{20})$ とすると式 (6.12) と式 (6.13) から

$$\boldsymbol{A}\begin{pmatrix}i & k\\ j & l\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \tag{6.14}$$

式 (6.14) が成り立ち、これを変形すると以下のようになる。

$$\begin{aligned}\begin{pmatrix}i & k\\ j & l\end{pmatrix} &= \boldsymbol{A}^{-1}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\\ &= \boldsymbol{A}^{-1}\end{aligned} \tag{6.15}$$

以上から i, j, k, l を求めることができる。さらに、式 (6.3) において $\boldsymbol{P}_e = \boldsymbol{P}_0, (a_e, b_e) = \boldsymbol{T}_0$ として $\boldsymbol{P}_c$ を作成し、$\boldsymbol{U}_p, \boldsymbol{V}_p$ と合わせて利用する。

6.4　入射球推定

ポリゴン内部に仮想的に配置した球のうち、どの球と視線が交わるかを求める。フラグメントシェーダでは球と視線との交点を直接与えられず、ポリゴンと視線ベクトル $\boldsymbol{V}$ との交点のみが与えられる。すなわち、ポリゴン上の同じ点に視線が到達したとしても同一の球に入射するとは限らない。そのため、視線が入射する球を正確に推定する必要がある。

まず、視線ベクトルとポリゴンとの交点を $\boldsymbol{P}$ とし、その点のテクスチャ座標を $\boldsymbol{T} = (a, b)$ とすると 6.3.1 節の $\boldsymbol{P}_c$ を用いて

$$\boldsymbol{P} = \boldsymbol{P}_c + a\boldsymbol{U}_p + b\boldsymbol{V}_p \tag{6.16}$$

と書ける。ここで、a, b をそれぞれの値が正のとき

$$a' = \lfloor a \rfloor + 0.5 \tag{6.17}$$

$$b' = \lfloor b \rfloor + 0.5 \tag{6.18}$$

負のとき

$$a' = \lceil a \rceil - 0.5 \tag{6.19}$$

$$b' = \lceil b \rceil - 0.5 \tag{6.20}$$

以上のようにすると、テクスチャの格子の中心を示す $\boldsymbol{P}'$ は次式のようになり

$$\boldsymbol{P}' = \boldsymbol{P}_c + a'\boldsymbol{U}_p + b'\boldsymbol{V}_p \tag{6.21}$$

さらに球の半径 r および $\boldsymbol{P}$ における法線ベクトル $\boldsymbol{N}$ を用いて

$$\boldsymbol{S} = \boldsymbol{P}' - r\boldsymbol{N} \tag{6.22}$$

とすると $\boldsymbol{S}$ は近傍球の中心座標の推定値となる。推定した球と視線ベクトル $\boldsymbol{V}$ との交差判定を行い、交差する場合には次の屈折計算へ進む。しかし、この操作を一度のみ行うだけでは不十分であり正確な近傍球を推定できておらず不具合を生じてしまう。間違った近傍球推定を行ってしまうと (Fig.**??**) のように、表示されるべき球の一部が消失してしまう。これを防止するために仮想的なスライス平面を作成し、オブジェクト内部方向へ段階的に移動させる (Fig.**??**)。球の半径 r よりも小さい値のオフセット距離を取り、法線の逆方向へポリゴンと平行に平面を順次作成していく。

視線ベクトル $\boldsymbol{V}$ を作成した仮想平面まで伸ばし両者の交点位置を求め、再度近傍球の推定を行う。球の推定にはこの交点位置におけるテクスチャ座標値が必要であるが、3 次元空間の点からテクスチャ座標値を逆計算するためには点がポリゴンの平面上に位置している必要がある。そのため、交点をポリゴンの平面上に投影した位置におけるテクスチャ座標値を再計算する。交点を $\boldsymbol{P}_s$ とすると、これをポリゴンの平面上に戻すためには h を仮想平面とポリゴンとの距離として

$$\boldsymbol{P}'_s = \boldsymbol{P}_s + h\boldsymbol{N} \tag{6.23}$$

とすればよい。$\boldsymbol{P}_s$ はポリゴン平面上の点なので

$$\boldsymbol{P}'_s = \boldsymbol{P}_c + a_s\boldsymbol{U}_p + b_s\boldsymbol{V}_p \tag{6.24}$$

以上の式が成り立つ。未知数 (a_s, b_s) は最小二乗法を用いて計算を行うと以下のように求めることができる。

$$\boldsymbol{P}_{s'c} = \boldsymbol{P}'_s - \boldsymbol{P}_c \tag{6.25}$$

$$a_s = \frac{(\boldsymbol{V}_p \cdot \boldsymbol{V}_p)(\boldsymbol{P}_{s'c} \cdot \boldsymbol{U}_p) - (\boldsymbol{V}_p \cdot \boldsymbol{U}_p)(\boldsymbol{P}_{s'c} \cdot \boldsymbol{V}_p)}{(\boldsymbol{U}_p \cdot \boldsymbol{U}_p)(\boldsymbol{V}_p \cdot \boldsymbol{V}_p) - (\boldsymbol{U}_p \cdot \boldsymbol{V}_p)(\boldsymbol{V}_p \cdot \boldsymbol{U}_p)} \tag{6.26}$$

$$b_s = \frac{(\boldsymbol{U}_p \cdot \boldsymbol{U}_p)(\boldsymbol{P}_{s'c} \cdot \boldsymbol{V}_p) - (\boldsymbol{U}_p \cdot \boldsymbol{V}_p)(\boldsymbol{P}_{s'c} \cdot \boldsymbol{U}_p)}{(\boldsymbol{U}_p \cdot \boldsymbol{U}_p)(\boldsymbol{V}_p \cdot \boldsymbol{V}_p) - (\boldsymbol{U}_p \cdot \boldsymbol{V}_p)(\boldsymbol{V}_p \cdot \boldsymbol{U}_p)} \tag{6.27}$$

求めた a_s, b_s をもとに、式 (6.17) から式 (6.22) までの操作を行い、再度交差判定を行う。最初に交差した球を第一の入射球として扱い、球の推定探索を終了する。交差していなければこの仮想平面による処理を仮想平面とポリゴンとの距離が r になるまで行い、交差する球があれば 6.5 節の処理へ進み、最後まで交差する球がなければ次の 6.6 節の処理へ進む。

6.5　屈折計算処理

球と交差した視線は屈折によって進行方向を変化させる。視線が入射した球の中心点を C とすると、点 C の座標値 $\boldsymbol{C}$ と視線ベクトル $\boldsymbol{V}$ および r から球と視線の交点 D が求まる (Fig.**??**)。点 D の座標値を $\boldsymbol{D}$ とすると、これらから点 D における法線ベクトル $\boldsymbol{N}_D$ は

$$\boldsymbol{N}_D = \frac{\boldsymbol{D} - \boldsymbol{C}}{\|\boldsymbol{D} - \boldsymbol{C}\|} \tag{6.28}$$

として正規化された値が得られる。この値と相対屈折率および視線ベクトル $\boldsymbol{V}$ からスネルの法則を用いて球の内部へ進入する屈折ベクトルを求めることができる。また、球から出射される屈折ベクトルについても同様の計算によって出射位置および方向がわかる。

****繰り返される屈折について可能であれば加筆****

6.6　テクスチャ座標再計算

レンズの役割を担っている球から出射されたベクトルは隣接する球と交差しなければ最終的にテクスチャ平面（6.1.1 節）へと到達する。テクスチャ平面上の到達点を $\boldsymbol{P}_T$ とすると、この点におけるテクスチャ座標値は 6.4 節と同様にポリゴンへの投影位置から求めることができる。

$$\boldsymbol{P}'_T = \boldsymbol{P}_T + 2r\boldsymbol{N} \tag{6.29}$$

座標 $\boldsymbol{P}'_T$ はポリゴン平面上にあるため式 (6.24) と同様に表すことができる。すなわち、求める座標は式 (6.26) および式 (6.27) のようにして求めることができる。

テクスチャ平面は色素細胞による色への寄与を求めるための平面であり、具体的にはテクスチャ平面上の視線の到達点の位置における色を返す。正確には、テクスチャ平面の色は複眼表面に現れる色そのものではなく、光の吸収度を代表するものと考えられる。色素細胞は昆虫などの複眼を持つ生物のなかで光を受容する部位に相当するので、光の吸収度が高いほど観測者にとっては暗く見える。そのため、適用されるテクスチャ画像はグレースケール画像であればよい。最終的には、画素値を正規化し通常のレンダリングにより計算した色へ乗算される。

6.7　光源処理および陰影処理

光源処理及び陰影処理は通常の Phong の反射モデル等を用いて計算を行う。正規化されたテクスチャ色情報を $\boldsymbol{c}_t$ とすると、最終的に描画される色 $\boldsymbol{c}$ は次の用に計算される。

$$\boldsymbol{c} = \boldsymbol{c}_t(\boldsymbol{a} + \boldsymbol{d}) + \boldsymbol{s} \tag{6.30}$$

ここで、$\boldsymbol{a}$ は環境反射光、$\boldsymbol{d}$ は拡散反射光、そして $\boldsymbol{s}$ は鏡面反射光である。

****頑張ってフレネル反射いれまっしょい！！****

第 7 章

結果と考察

7.1　＊サブセクション＊

さぶせく～～。

第 8 章

結論

8.1　結論

本研究では、複眼のリアルタイムレンダリングを行った。などなど。以下の成果を確認できた。

-
-
-
-

本研究は～だけではなく……。

8.2　今後の展望

第 8.1 章で既述したように……。といった使い方ができる。

謝辞

　あとは謝辞をつらつらと書くだけよん♪謝辞

　謝辞謝辞謝辞謝辞謝辞謝辞謝辞謝辞謝辞謝辞謝辞

　謝辞変更ローカルで。

以上

1 p ～ 28 p 完

修士学位論文

平成 26 年度

東京大学大学院 学際情報学府先端表現情報学コース

136313 佐川 和輝