

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

第 1 章 序論	1
1.1 研究背景	1
1.1.1 体積や厚みを考慮したコンピュータグラフィックス	2
1.1.2 生物発想のコンピュータグラフィックス	3
1.1.3 リアルタイムでのハイクオリティレンダリング	4
1.1.4 複眼と偽瞳孔	4
1.2 本研究の目的	5
1.3 本論文の構成	5
第 2 章 関連研究	7
2.1 微細構造を扱った研究	7
2.1.1 マイクロファセットモデル	7
2.1.2 構造色 CG	9
2.1.3 織布のレンダリング	12
2.2 本研究の位置づけ	13
第 3 章 複眼の光学的特徴	15
3.1 複眼の性質	15

3.1.1	連立像眼	16
3.1.2	重複像眼	16
3.2	個眼のしくみ	16
3.3	偽瞳孔	17
3.3.1	Central Pupil	19
3.3.2	Side Pupil	19
第 4 章	予備実験	23
4.1	実験の目的	23
4.2	実験方法	24
4.3	結果と議論	26
4.3.1	ビー玉がひとつの場合	26
4.3.2	木枠を用いてビー玉を密集させた場合	26
4.3.3	ビー玉とパンチングメタルの穴の配置を合わせた場合	28
4.4	まとめ	31
第 5 章	提案手法	35
5.1	フレームワーク	35
5.1.1	複眼の光学モデル	35
5.1.2	アルゴリズムの全体像	36
5.2	テクスチャと球の配列	37
5.3	オブジェクトファイルデータから計算用データの生成	38
5.3.1	テクスチャ座標軸方向 3 次元単位ベクトル	40
5.4	入射球推定	45

5.5	屈折計算処理	48
5.5.1	隣接球処理	48
5.6	テクスチャ座標再計算	51
5.7	光源処理および陰影処理	51
第 6 章 結果と考察		53
6.1	実装方法および実行環境	53
6.2	処理系	54
6.2.1	内部通過処理および外部通過処理	54
6.3	ジオメトリ	55
6.3.1	レンズ半径と偽瞳孔の数	55
6.3.2	歪みとテクスチャの関係	57
6.3.3	さまざまなテクスチャ	59
6.4	パラメータと実行速度	60
6.4.1	隣接球の処理 ON/OFF	62
6.4.2	視点との距離による変化	70
6.4.3	個眼密度	74
6.5	実物との比較	75
6.6	さまざまなオブジェクト	79
6.7	応用事例	82
第 7 章 おわりに		85
7.1	結論	85
7.2	今後の展望	86

謝辞

87

参考文献

89

図目次

2.1	一つめの図	8
2.2	二つめの図	8
2.3	一つめの図	8
2.4	二つめの図	8
2.5	ハイヒールとオーナメントの絵 []	9
2.6	シャボン玉の薄膜干渉 []	10
2.7	鉱石の多層膜干渉 []	10
2.8	CAPTION	12
2.9	一つめの図	13
2.10	二つめの図	13
2.11	クッション	14
3.1	連立像眼のしくみ	17
3.2	重複像眼のしくみ	18
3.3	Side pupil の原理	21
4.1	実験の材料	24
4.2	パンチングメタルを画用紙に固定	25

4.3	ひとつのビー玉による像	27
4.4	木枠による密集	28
4.5	6つの球による穴の拡大像	29
4.6	穴にそって整列されたビー玉	29
4.7	視点を装置と平行に動かした場合	30
4.8	黒点の大きさの違い	31
4.9	視円錐と黒点の大きさ	32
5.1	シェーディングモデル	36
5.2	アルゴリズムフレームワーク	37
5.3	テクスチャと球の位置関係	38
5.4	テクスチャおよび球の格子状配置	39
5.5	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	42
5.6	平面上の変位ベクトル	43
5.7	隣接球処理アルゴリズム	49
5.8	領域分割	50
6.1	外部通過処理に見られるノイズ	54
6.2	レンズ半径 r と偽瞳孔の数の関係	56
6.3	テクスチャの継ぎ目	58
6.4	基本テクスチャと適用結果	60
6.5	テクスチャの変更による外観の変化	61
6.6	評価オブジェクト（ワイヤフレーム表示）	62
6.7	上限値による画像の変化 1	64

6.8	上限値による画像の変化 2	65
6.9	各偽瞳孔の名称および場所	67
6.10	実行速度と l の関係	68
6.11	上限値による画像の変化比較 3	69
6.12	上限値による画像の変化比較 4	70
6.13	視点距離と偽瞳孔の位置関係（平面オブジェクト）	71
6.14	視点距離と偽瞳孔の位置関係（球体オブジェクト）	72
6.15	視点距離と実行速度の関係	73
6.16	視点距離と実行速度（平面）	74
6.17	視点距離と実行速度（球体）	75
6.18	個眼密度による実行結果の変化（平面）	76
6.19	個眼密度による実行結果の変化（球体）	77
6.20	個眼配置と偽瞳孔の関係	78
6.21	実物の side pupil	79
6.22	六角形の偽瞳孔（写真:[4] より転載）	80
6.23	リング状の偽瞳孔（写真:[4] より転載）	81
6.24	貝殻オブジェクトへの適用	82
6.25	鳥頭オブジェクトへの適用	83

表目次

6.1	実行環境	53
6.2	オブジェクト情報	62

第 1 章

序論

1.1 研究背景

フォトリアルなコンピュータグラフィックス (Computer Graphics: CG) は、ハードウェアの発展や高性能コンピュータの普及により、近年ではさらに身近なものになっている。実物と見間違うほどの CG も珍しくなくなり、観客は実物と CG との差異に敏感になってきている。CG 技術の発展により、コンピュータによって作成された画像に対し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。CG の歴史上においても、人間が現実世界で目にするのをそのままコンピュータ上に再現することが当初からの目標とされており、一部では実物と CG との差異が埋まりつつあると言える。

成熟期に入ったとも考えられる CG 技術ではあるが、リアリズムの追求はとどまるところを知らず、さらなる技術向上が目指されている。また、現実世界には CG によって十分に再現されていないさまざまな光学的な現象が依然として存在することも事実である。そのため、多くの研究者たちが CG の表現力をさらに向上させるために研究・開発を行っている。

1.1.1 体積や厚みを考慮したコンピュータグラフィックス

CGは物体がどのように「見えるか」をコンピュータによって再現するために生み出された技術である。そのため、「ものの見た目」においてもっとも重要である物体表面での光の反射がまず取り扱われる。しかしながら、実際には臓器などの不透明な物体や煙や炎などといった粒子で満たされた一定の領域や体積を考慮する必要のある対象も存在する。こうした対象を視覚化(visualize:ビジュアライズ)するためには、光が物体内部や粒子で満たされた領域内部を通過することによってその挙動にどのような影響が及ぼされるかについても考慮する必要がある。

たとえば、ボリュームレンダリング(Volume Rendering)の手法では、空間をボクセル(voxel)と呼ばれる六面体に分割し、遮る光の量を計算する。ボクセルごとの密度も考慮しており、密度の大きなボクセルほど多くの光の量を遮る。歴史的には、医療の分野で用いられるCT(Computed Tomography)データのビジュアライゼーションを目的とする視覚化手法の研究が重ねられていたが^[1]、1980年代にCGの分野でもボクセルを用いたボリュームデータの視覚化が行われるようになった^[2]。ボリュームレンダリングは主に肉や骨、脂肪といった密度の違う領域をあわせ持つ人体の視覚化する方法として研究が始められたようである。

ほかには、パーティシペイティングメディア(participating media)という自然科学の分野における考え方を導入した手法がある。煙や炎などは光が物体内部で吸収や散乱を起こしやすく、こうした光の吸収度や散乱度が非常に大きい領域はパーティシペイティングメディアと呼ばれる。パーティシペイティングメディアをレンダリングするための技法はボリュームレンダリングの技法とともにボリュームをレンダリングするための技法として、CGの歴史上でほぼ同時期に登場した^[3]。

大理石や人体の皮膚といった半透明の材質をレンダリングするための技法として、サブサー

フェイススキャッタリング (subsurface scattering) を考慮した手法も研究が盛んに行われている。サブサーフェイススキャッタリングとは、一度物体内部に進入した光が物体内部で反射を繰り返し、光が入射した位置とは別の位置から外部へ向かって出射する光の挙動のことである。自然界のほとんどの物体でこの現象が見られ、半透明の質感を持った大理石や皮膚などにおいて顕著に現れる。サブサーフェイススキャッタリングを物理的に正確にレンダリングするためには非常に大きな計算不可がかかっていたものの、映画などのフォトリアリスティックな表現への要求から研究が進み、現在ではすでに実時間での計算手法も開発されている [1]。

以上のように、光が物体内部を通過するオブジェクトのレンダリングに関しては以前から研究が進んでおり成果を上げている。しかし、いずれも物質レベルのミクロな構造に対する光の挙動を再現したものであり、ミドルレベルの大きさの構造体が集合した物体に対する光の挙動までは扱っていない。

1.1.2 生物発想のコンピュータグラフィックス

皮膚や毛髪など人間を対象とした CG 表現に関してはこれまでに数多くの研究がなされているが [2]、人間以外にも動物や昆虫といったさまざまな生物のもつ光学現象を対象とした CG 研究も近年では行われるようになってきている。生物の発する体色などの色は発色のしくみが解明されていないものもあり、たとえば構造色といった表面微細構造によって色が変化するものも存在する。構造色は光の散乱、回折、干渉などのさまざまな光学現象によって生じる色であり、物体表面の膜構造や回折格子構造などに由来する。そのため、通常の色素による発色とは異なり、角度や媒質などの観測条件によって色が変化するものが多い。自然界ではコガネムシやモルフォチョウの翅などの昆虫の体色や孔雀の羽根などが該当する。現在盛んに行われている物理学を CG に応用した研究と比較して、生物のモーションや体色などの生物学を CG に応用した研究にはまだ手がつけられていない分野も多いため、今後の発展が期待される。

1.1.3 リアルタイムでのハイクオリティレンダリング

近年ではコンピュータ・ハードウェアの高速化により、リアルタイム (real-time:実時間) における CG 映像の表現力が飛躍的に向上している。リアルタイムで実写に迫る表現も可能になりました。今後もその要求は高まると考えられる。*****

1.1.4 複眼と偽瞳孔

複雑な内部構造を有し、さらに生物に関連するものとして昆虫や甲殻類などの複眼が挙げられる。複眼は個眼と呼ばれる小さな眼の集合体であり、それぞれの個眼にはレンズや色素細胞といった光学現象に影響する部位が備わっている。複眼の見た目に関係する光学的な現象として偽瞳孔と呼ばれる現象がある。偽瞳孔とは複眼表面に現れる黒い斑点模様であり、個眼のレンズによる光の屈折作用と密接に関わっている。

複眼を構成する個眼の数は非常に多く、さらに表面の構造が非常に複雑で細かいためコンピュータで精巧な形状を作成するのは現実的ではない。また、精巧な形状のみを擬似的に表現する手法は CG においても研究が進んでおり、非常に高い効果を上げている。例を挙げると、ディスプレイスメントマッピングやバンプマッピングなどの手法では、ポリゴン表面に変位情報や法線情報を付加するによって複雑で細かい表面形状を少ない計算量で表現することを可能にしている (Fig.??)(Fig.??)。現在ではアートやエンターテイメントの分野において昆虫の複眼を表現する際にこれらの手法を利用するのが一般的となっている (Fig.??)。しかしながら、上記の手法では複眼の表面的な形状を再現するのみにとどまっており、複眼の内部構造を考慮した光学的現象を再現するまでには至っていない。すなわち、先述の偽瞳孔を CG を用いて再現するための手法は未発展であると言える。

1.2 本研究の目的

これまでに紹介したように、技術の進歩とともに、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、今後はリアルタイムレンダリングにおいても高精細で写実的な表現が求められると考えられる。1.1.1 節で述べたように、物体内部を通過する光を考査したレンダリング（rendering）手法が生み出されてはいるものの、より複雑で光の挙動に影響を与えるような内部構造を持つ物体のレンダリングに関しては対象ごとに個別の手法を考案する必要がある。本研究では、昆虫などの複眼表面に現れる光学現象として 1.1.4 節で述べた偽瞳孔に着目し、この現象の CG によるリアルタイムでの表現手法の研究および開発を目的とする。

1.3 本論文の構成

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 3 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、過去の研究を踏まえ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 5 章では、本研究で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

本研究では、複眼の偽瞳孔現象を対象としている。偽瞳孔は複眼表面の微細構造によって巨視的に現れる模様であり、偽瞳孔をコンピュータグラフィックスの分野で扱った研究は未だ存在しない。しかしながら、複眼同様に表面の微細構造を有している材質などのレンダリングを扱った研究は少なからず存在するため、それらを紹介し、本研究との比較を行う。

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。本節では、マイクロファセットモデル、構造色、織布のレンダリングに関する研究をそれぞれ簡略して紹介する。

2.1.1 マイクロファセットモデル

表面のマイクロファセット（microfacet）構造の例として車の塗装のようにざらついたようなきらめきを持つ材質がある。こうした材質をレンダリングする手法として、Günther ら [] はノーマルマップ（normal map: 法線マップ）を利用し、Rump ら [2] は測定したテクスチャを

用いた手法をそれぞれ提唱している。



Fig. 2.1 一つめの図



Fig. 2.2 二つめの図



Fig. 2.3 一つめの図



Fig. 2.4 二つめの図

また、Wenzel ら [] の行ったハイヒールの表面やクリスマスオーナメントのようにきらめく材質のレンダリングに関する研究では、サーフェスもしくは観測者の動きに合わせて変化するきらめきのランダムなパターンを表現している。この手法の対象は鏡のような薄片を含むダイナミックにきらめく表面材質、およびかすかに小さいスケールのきらめきを示す粗い表面材質である。これらの現象は原則的には Günther らのように高解像度のノーマルマップによって表現

することができる。しかし、細かな特徴をもつマップは角度のついた照明条件下ではエイリアシング (aliasing) において重大な問題を抱えてしまう。それゆえ、Wenzel らは通常では連続しているマイクロファセットの分布をサーフェス上の離散的な散乱粒子の分布と置き換えた、確率論的なモデルを提唱している。この確率論的な階層では、個別の粒子を考えることなく多数のランダムな粒子の存在下で効率的な評価を行うことができ、それによってマルチスケールにおいて双方向反射率分布関数 (BRDF: Bidirectional Reflectance Distribution Function) を導くことができる。



Fig. 2.5 ハイヒールとオーナメントの絵 []

2.1.2 構造色 CG

普段、人間が知覚する色の多くは色素によるものであり、光の吸収や反射によって色として認識される。一方、構造色は光の吸収が電子によるものでなく、光の干渉による強めあいや弱めあい、光の散乱などによって起こる現象 [] である。そのため、媒質の屈折率が変化すると色素による色は変化しないが構造色は変化するなどの特徴がある。

こうした性質のある構造色を CG で表現する研究は 1980 年代にはすでに行われており、現在に至るまでさまざまな物質や気象現象に関する構造色が CG によって表現されている。そ

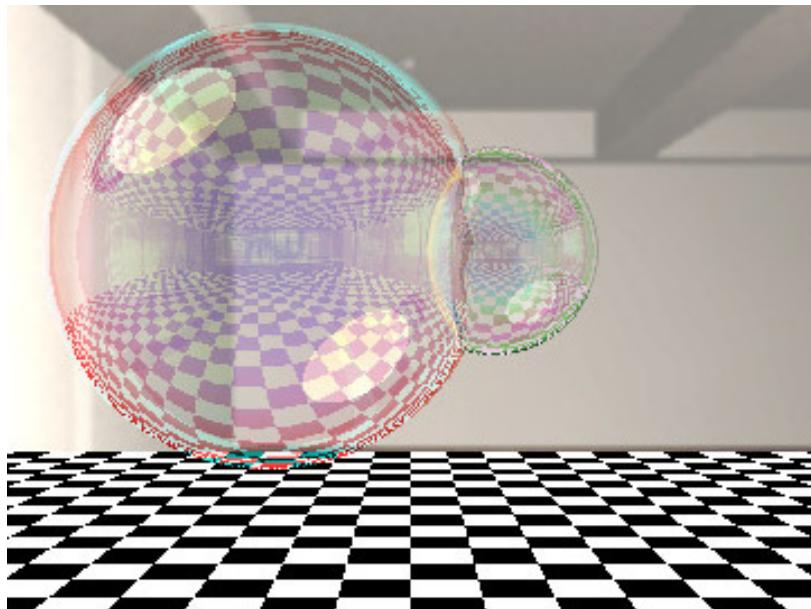


Fig. 2.6 シャボン玉の薄膜干渉 []

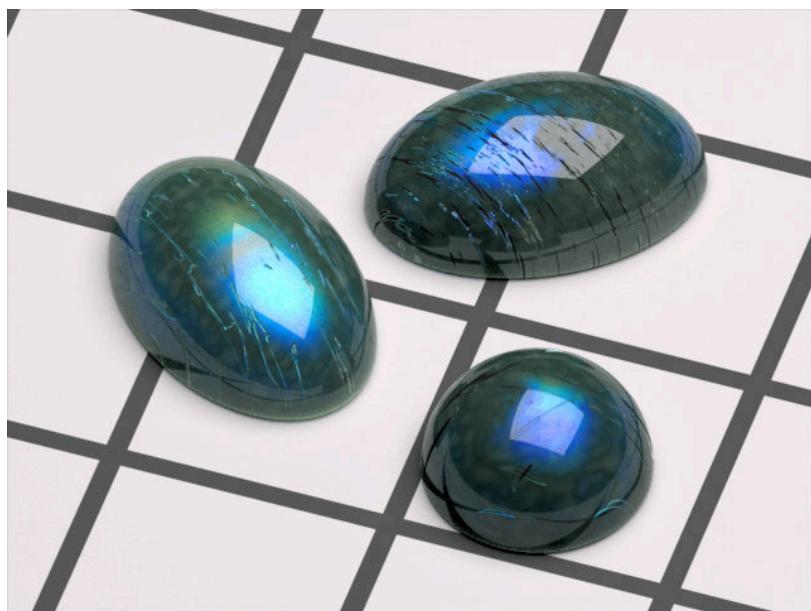
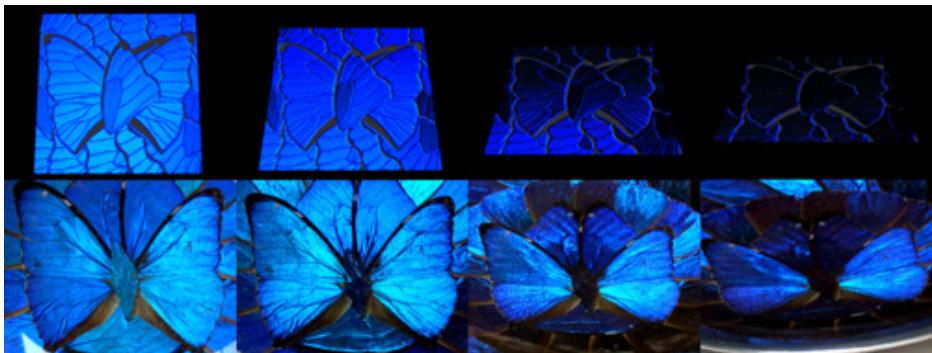


Fig. 2.7 鉱石の多層膜干渉 []

の中でも、とくに物体の表面構造が関係している構造色の例としては、薄膜干渉によるシャボン玉 (Fig.2.6) や多層膜干渉による鉱石 (Fig.2.7)、回折によって生じる CD などの光学ディスク (Fig.??)、そのほかにも鱗粉のミクロ構造によるモルフォチョウの翅 (Fig.??) などが挙げられる。

シャボン玉の表面は二重膜の薄膜構造になっており厚みが光の波長以下となるため、薄膜の表面で反射する光と薄膜の裏面で反射する光が干渉を起こし虹色に見える。Icart ら [] は重力による薄膜の厚みの変化を考慮したシャボン玉の CG 表現の研究を行い、岩崎ら [] は薄膜の厚みが動的に変化するシャボン玉をレンダリングする研究を行い、事前計算によるテクスチャを生成することによってリアルタイムでの表現を可能とした。これらのように、形状などのジオメトリの変化が色に変化を与えることも構造色の特徴である。また、多層膜のように干渉する膜の数が増えると、光の計算はさらに複雑になる。

****回折、モルフォの解説****



(a) CAPTIONa[]



(b) CAPTIONb[]



(c) CAPTIONc[]

Fig. 2.8 CAPTION

2.1.3 織布のレンダリング

織布は織糸の小さいスケールの立体構造によって変化するさまざまな外観を有する。その構造的なディティールを正確にモデリングすることで非常にリアリスティックな織布のレンダリングをすることができるが、織糸レベルのボリューメトリック (volumetric) なモデルを作成するには非常にコストがかかる。

Zhao ら [1] は織糸の立体構造を自由に構築するために、詳細な生地サンプルをスキャンするのではなく、シンプルな織り構造の生地サンプルをスキャンすることによってボリュームトリックな標本のデータベースを作成し、それらからデータをコピーすることによって目的のボリュームデータを合成している。その結果、統一的にさまざまな織りの布を表現することができ、大小のスケールの両方で非常にリアルな出力画像を生み出すことができている。

****もう少し解説を追加****



Fig. 2.9 一つめの図



Fig. 2.10 二つめの図

2.2 本研究の位置づけ

先述のように、表面微細構造を考慮したレンダリングに関する研究はいくつか存在する。それらは本研究で対象としている複眼の CG 表現と直接的に関連する研究ではないが、いずれも微細な構造が巨視的な外観に影響する材質に関する研究であると言える。マイクロファセットおよび構造色は表面構造を持つといえども、全体としては均一な表面材質である。一方、織布のレンダリングに関しては、肉眼で目視可能なレベルのスケールの構造物があり、複眼と同様に一定の条件で整列した構造をしている。



Fig. 2.11 クッション

****比較とか**** ほかのやつはたいていランダムにやってもできるけど、本研究の対象である複眼は綺麗に並んでるやつだからそう簡単にはいかないよ

第3章

複眼の光学的特徴

本章では、昆虫や甲殻類に見られる複眼に関する知識や研究を紹介する。

3.1 複眼の性質

複眼は個眼と呼ばれる単位の視覚器官から構成される。ひとつの個眼にはレンズと光受容細胞 (photoreceptor cell) および色素細胞 (pigment cell) などが含まれる。一般に、複眼を構成する個眼の数は複眼の大きさに比例し、少ないものでは数百個、トンボなどの多いものでは2万個以上であると言われている。

八木[]によると、昆虫に見られる複眼には複数のタイプがあり、真円錐眼 (eucone eye)、偽円錐眼 (pseudocone eye)、無円錐眼 (acone eye)、外円錐眼 (exocone eye) の4種類に大別される。真円錐眼は堅いキチン質の円錐晶体をもち、独特な光学的性質を有している。偽円錐眼は流動体もしくは半流動体で満たされたカプセルから構成され、比較的弱い屈折力しか持ち合っていない。無円錐眼は円錐晶体を有しておらず、外円錐眼では円錐晶体が角膜面のくぼみによって置き換えられている。真円錐眼はさらに連立像眼と重複像眼に分けられる。

3.1.1 連立像眼

チョウなどの眼は連立像眼 (apposition eye) と呼ばれ、昼行性の種に多く見られる。感覚神経のユニットは円錐晶体の頂点から離れ、ひとつずつ独立して光を受容するような構造になっている []。すなわち、ひとつひとつの個眼は光学的に独立しており、レンズ直下の感桿に導かれた光は外に漏れることなく感桿基部まで到達する []。Nilsson[] によると連立像眼はさらに少なくとも 4 タイプに分類され、単純連立型、無限焦点型 (afocal type)、分散感桿型 (open rhabdom)・神経重複型 (nural superposition)、透明連立型などがある。単純連立像眼と無限焦点連立像眼の基本構造はほぼ同じであるが、無限焦点連立像眼では単純連立像眼とは異なり感桿へ平行光が入射する。蟻川 [] によると、単純連立型はハチ類やカニ類に多く見られ、無限焦点型はチョウ類で発見されたタイプであると言われている。また、分散感桿型・神経重複型はハエ類などの一部の昆虫に見られ、透明連立型は浮遊性の甲殻類にのみ発見されている特殊なタイプであると言われている。

3.1.2 重複像眼

連立像眼が昼行性の種に多く見られるのに対して、重複像眼はガなどの夜行性の種に多く見られる。重複像眼の特徴としては、色素細胞の色素粒子が光の明暗によって上下に移動する点が挙げられる。重複像眼の色素細胞は暗いところでは円錐晶体を取り囲むように上部へ移動する。夜行性の昆虫などの眼が真っ黒に見えることと関連があると考えられる。

3.2 個眼のしくみ

一般に、複眼が多数の小さな個眼の集合体であり、半球状のドームを形成していることはよく知られている。個眼の表面は凸レンズ構造をしており、それぞれの形状は六角形もしくは正

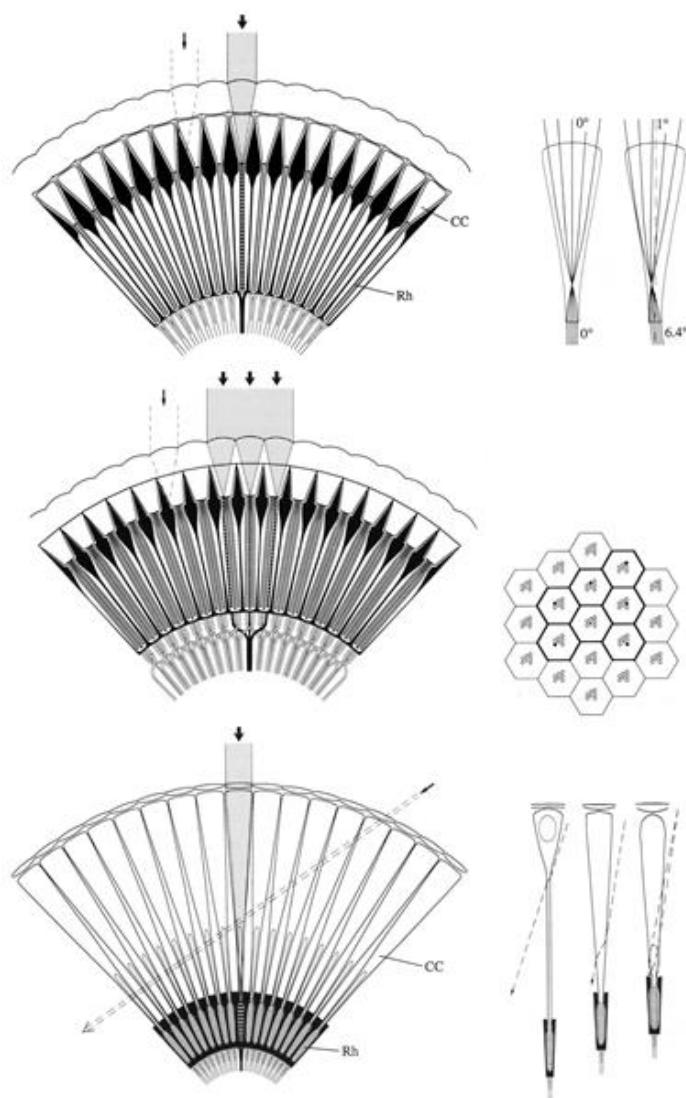


Fig. 3.1 連立像眼のしくみ

方形である (Fig.??)。個眼のひとつひとつは主にレンズ、色素細胞、円錐晶体そして感桿などから構成されており、複眼の偽瞳孔に関係が深く、外部的な見た目に影響する部位はレンズおよび色素細胞である。

3.3 偽瞳孔

1.2 節で述べた偽瞳孔についてより詳細な説明を行う。偽瞳孔は複眼表面に見られる暗い斑点状の模様であり、チョウやトンボ、セミなどさまざまな昆虫で顕著に現れる。また、昆虫だ

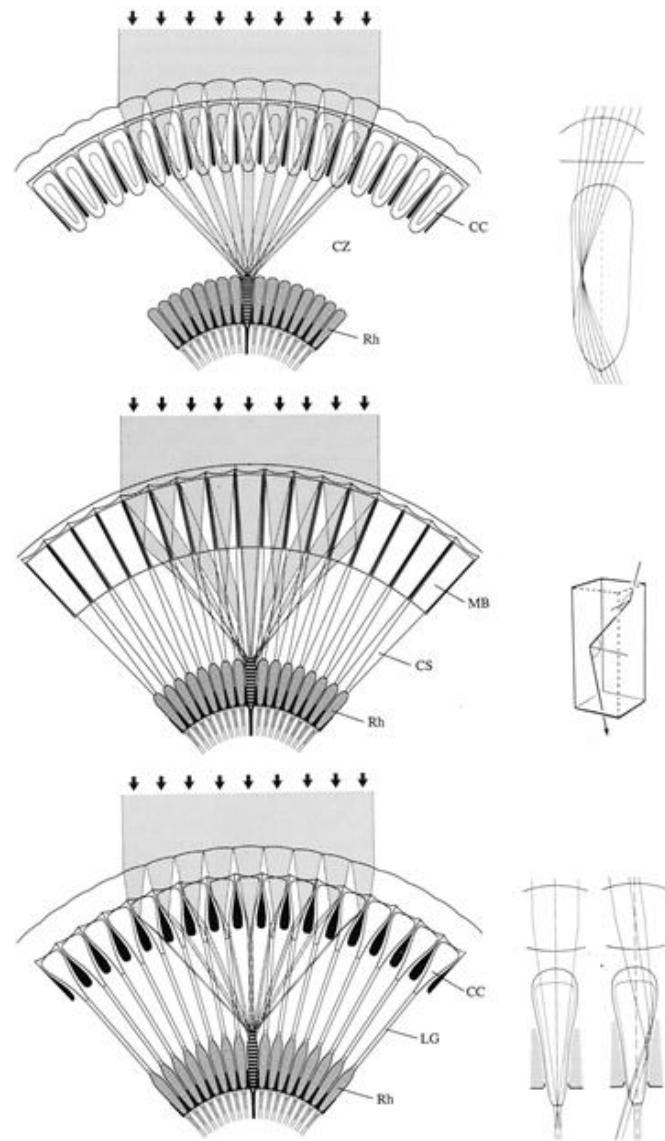


Fig. 3.2 重複像眼のしくみ

けではなくエビやカニなどの複眼を持つ甲殻類にも見られる。

初めに偽瞳孔の現象を発見したのは Leydig[] であり、Leydig は *Limulus* の複眼において瞳孔のような暗い斑点を観測し、脊椎動物の瞳孔とは違って外部から観察する方向によって斑点の位置が変化することに言及している。

基本的には、偽瞳孔は中心に位置する暗点とそれを取り囲む 6 つの暗点から構成されている。周囲にある 6 つの暗点は中心の暗点を取り囲む六角形の頂点の位置に見られる。八木 []

は、これらの点のうち中心に位置する暗点を central pupil、周囲の位置にある 6 つの暗点を side pupil と呼称している。また、side pupil の外側を取り囲むように配置される暗点も存在し、ひとつめの side pupil を中心として先述と同様に六角形状に位置する。これらの暗点のうち、central pupil と side pupil をのぞいたものは second side pupil と呼称されている。本研究でもこれらの呼称をそのまま用いることにする。

3.3.1 Central Pupil

八木は色素細胞の位置によって central pupil の形状の変化が見られることについて記述している。色素細胞が円錐晶体の下端にある場合 (Fig.??)、central pupil は比較的丸い形状の単純な暗点として現れる。この場合、色素細胞が少し円錐晶体の上部へ移動した場合と比較すると大きさはやや小さい。色素細胞が少しだけ上部へ移動した場合には、偽瞳孔は少し大きくなり暗さを増す。また、色素細胞がさらに移動し、レンズの近くまで上がってきた場合には、偽瞳孔の形状が六角形に近づいていくという。

全ての偽瞳孔に当てはまるわけではないが、central pupil の中央には茶色もしくは赤みがかった明るい点を示すものがある。これは、感桿を通して基底膜から反射した光によるものである。しかしながら、この明点は一部の種をのぞいた他の種ではほとんど見られない。なぜなら、色素細胞が基底部で円錐晶体を囲っており基底膜への光の進入を抑えているからである。重複像眼では基底膜および気管のタペータム (tapetum:輝板) からの反射が合わさって赤もしくは琥珀色の明るい輝きになる。

3.3.2 Side Pupil

Side pupil は central pupil から同距離に位置し、色素細胞からの間接反射によって起こる。Central pupil は光が入射したレンズが属している個眼の色素細胞の色を反射しているが、side

pupil は光が入射したレンズが属している個眼に隣接する個眼の色素細胞からの反射によって生じる []。

光が (Fig.3.3) の A のような道筋を通る場合、この個眼の表面は central pupil の一部として観測される。一方、光が (Fig.3.3) の C のような道筋を通る場合、この個眼の表面は side pupil の一部として観測される。円錐晶体には色素細胞に覆われている部分と覆われていない部分がある。(Fig.3.3) の B および D の道筋を通る視線は、色素細胞に覆われていない円錐晶体の一部に到達している。通常、これらの位置にぶつかった光は円錐晶体の奥にある体液の色を反射し薄い色として観測される。

角度を変えながらひとつの個眼のレンズを観測すると、ある特定の角度の範囲内で円錐晶体の壁面に沿った色素細胞の色を反射することは明らかである。逆に、円錐晶体において色素細胞のない壁面の色を反射する角度の範囲も存在すると言える。すなわち、複眼表面において central pupil と side pupil の間には色素のない領域が存在することは想像に難くない。また、second side pupil の生じる原理も同様である。

偽瞳孔の暗点は central pupil, side pupil, second side pupil の順に色が薄く見えるものも存在する。その理由としては、side pupil および second side pupil は隣接する個眼を通過して色素細胞の色を反射するため、通過する個眼のレンズや体液が厚くなり、光が減衰するからであると推測される。

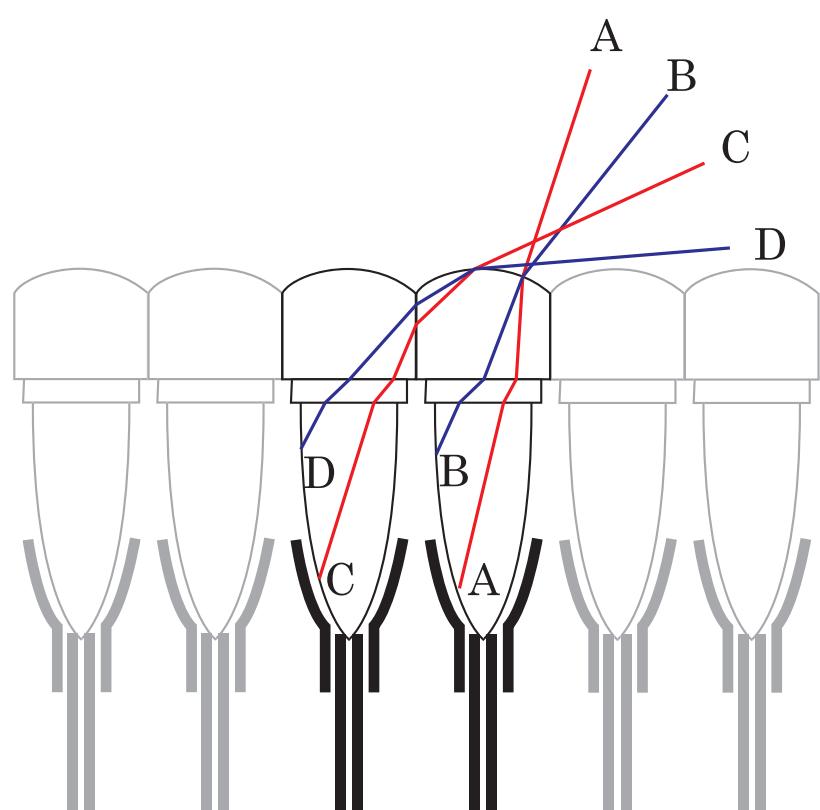


Fig. 3.3 Side pupil の原理

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 [5] 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

実験に用いた材料は以下のとおりである (Fig.4.1)。

- 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

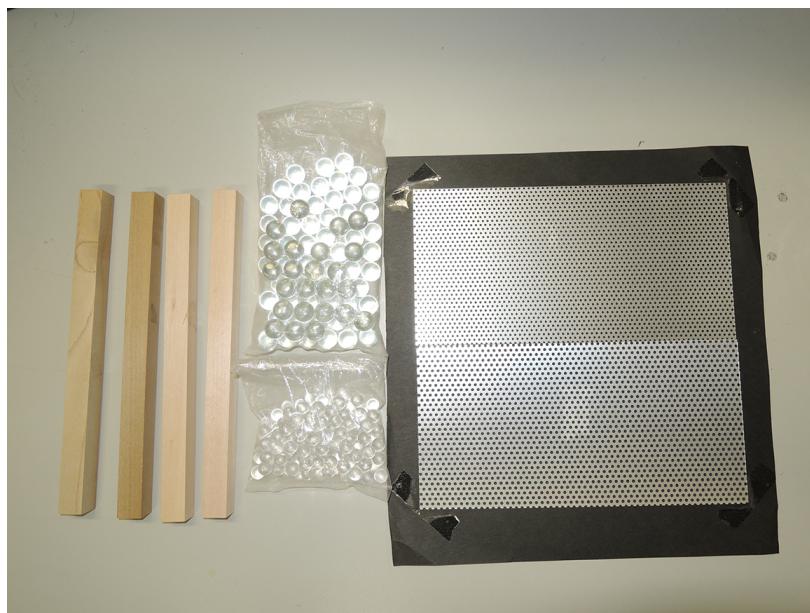


Fig. 4.1 実験の材料

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 (Fig.4.2)。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。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。続いて、ビー玉をパンチングメタルの上に配置し、球体レンズによる光の屈折がどのような像を生み出すのかを確認する。ビー玉をひとつのみ配置した場合、木枠により複数のビー玉を密集させた場合、そしてビー玉とパンチングメタルの穴の配置を合わせた場合のそれぞれについてビー玉に写る像の観察を行った。木枠を用いる場合では、まず木の棒で枠を作つてパンチングメタルの上に乗せ、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。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、木枠を用いる場合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。ビー玉とパンチングメタルの穴の配置を合わせた場合では、パンチングメタルの穴の直上に必ずビー玉が乗るようにし、複数のビー玉が整列するようになっている。



Fig. 4.2 パンチングメタルを画用紙に固定

4.3 結果と議論

使用したパンチングメタルの大きさは、穴同士の距離が約3.0mmで穴の径が約1.0mmのものと穴同士の距離が約3.5mmで穴の径が約1.5mmのもの。ビー玉は径が約12.0mmおよび約8.0mmのものを使用した。木の棒はそれぞれ幅15.0mm、高さ15.0mm、奥行き200.0mmのもの4本を加工して木の枠を作成した(Fig.4.1)。また、パンチングメタルの大きさとビー玉の大きさがそれぞれ2種類ずつあるため、これらを組み合わせたモデルに対して観察を行った。

4.3.1 ビー玉がひとつの場合

まずははじめに、ひとつのビー玉をパンチングメタルの穴の直上に置いて、ビー玉表面に写る像を観察した。パンチングメタルを正面に近い方向から見ると、黒点(パンチングメタルの穴)が拡大された虚像を確認することができた(Fig.4.3(a))。さらに、レンズを見る角度を変えたパンチングメタルと平行に近づけていくと、ある角度を境に倒立像が観測されるようになつた(Fig.4.3(c))。また、球の直下の黒点を像として写した場合よりも、隣接した黒点を写した際に球に占める黒い部分が大きくなるという結果が得られた(Fig.4.3(b))。すなわち、穴の正面からビー玉を見る場合よりもある程度角度をつけた場合のほうがビー玉全体が黒く見える。以上から、球の直下およびその周辺の黒点に対してビー玉が拡大レンズとして作用していることが確認できた。

4.3.2 木枠を用いてビー玉を密集させた場合

続いて、木枠を用いてビー玉を密集して配置した(Fig.4.4)。この場合、ビー玉の直下にパンチングメタルの穴があるとは限らないため、各ビー玉ごとに写る像の違いが大きいことに注意する。ビー玉の大きさを変えても特に大きな違いはなく、各パンチングメタルを拡大した虚像

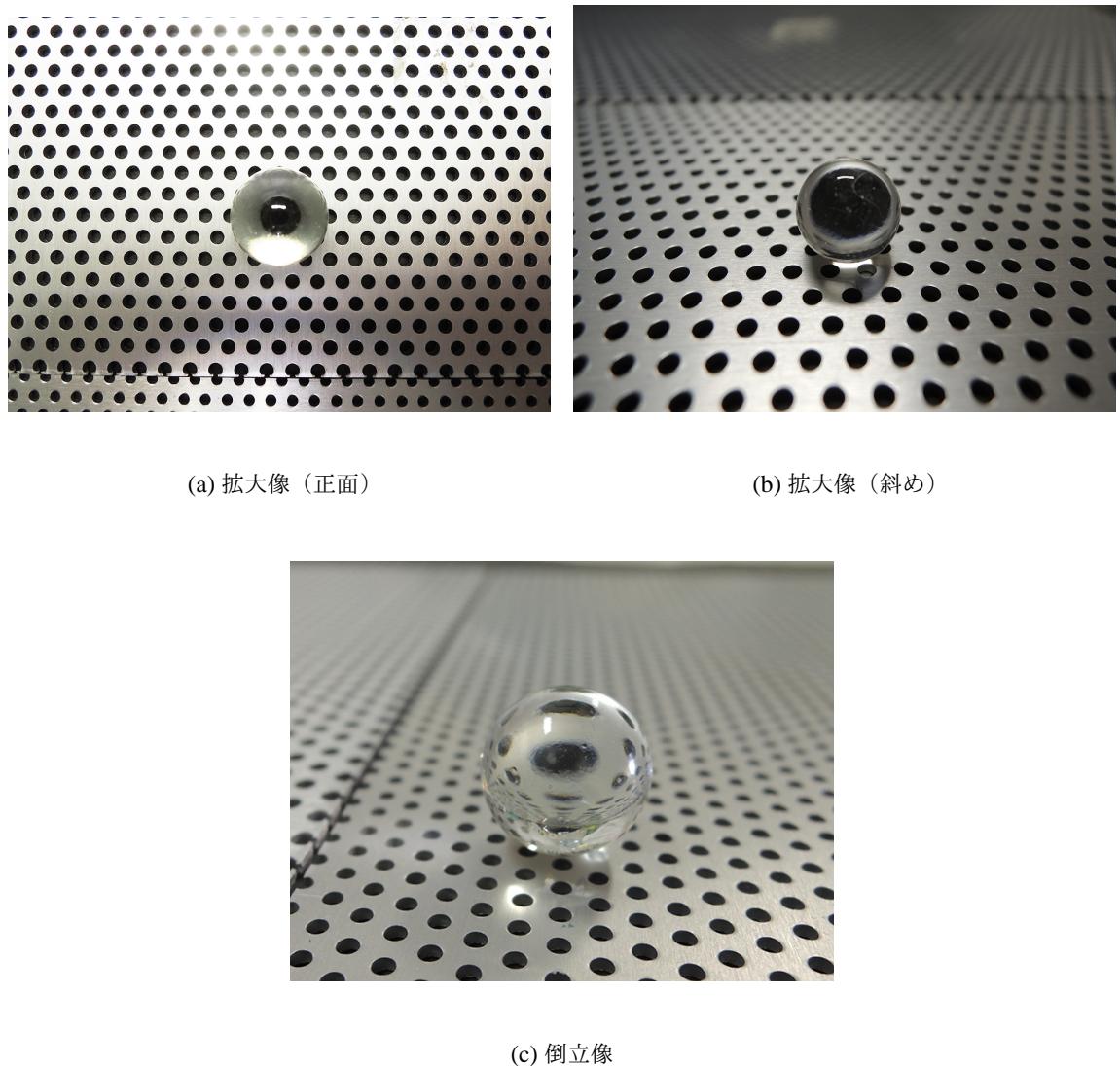


Fig. 4.3 ひとつのビー玉による像

が観測され、生じる像の周期はパンチングメタルの大きさによって変化した。パンチングメタル上のビー玉を木枠と共に動かすことによって、拡大された黒点が流れるように動く様子を肉眼で確認することができる。このことから、球体レンズを密集させることによって奥で接している面上の様子を拡大できることを確認した。

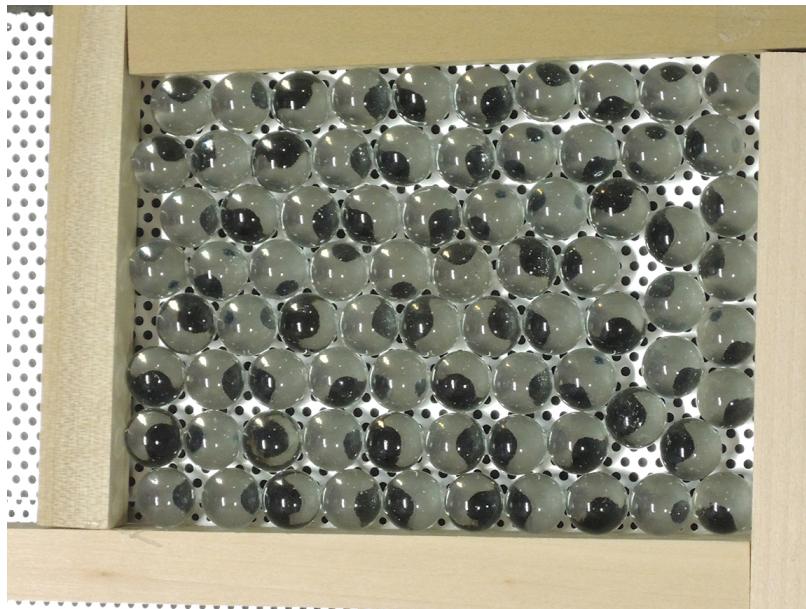


Fig. 4.4 木枠による密集

4.3.3 ビー玉とパンチングメタルの穴の配置を合わせた場合

前述の木枠を用いた方法ではビー玉の周期とパンチングメタルの穴の周期が一致しておらず、球の直下に必ずパンチングメタルの穴を配置することはできなかった。そこで、穴同士の距離と級の直径がおおむね一致する、径が約 12.0mm のビー玉および穴同士の距離が約 3.5mm で穴の径が約 1.5mm のパンチングメタルを用いて球の直下に必ず黒点が来るような配置を作成した (Fig.4.6)。

木枠を用いる方法と比較して、この方法では拡大された黒点をよりはっきりと確認することができる。その理由として、隣り合う球同士の条件の違いが光源および視点との位置関係のわずかな変化のみであるためだと推測できる。隣り合う球がお互いに似た像を写すのでノイズが少なく、黒点の拡大像によって全体的に黒くなった複数のビー玉によってひとつのまとまった黒い領域が形成されたり、黒点を写さない複数のビー玉によって広い無色の領域が形成されるのであろう (Fig.4.7)。

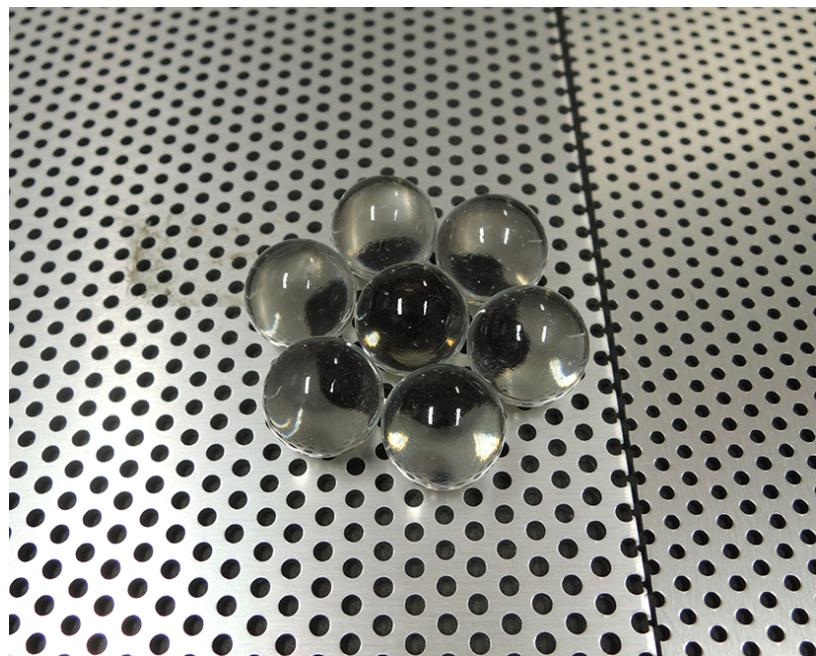


Fig. 4.5 6つの球による穴の拡大像

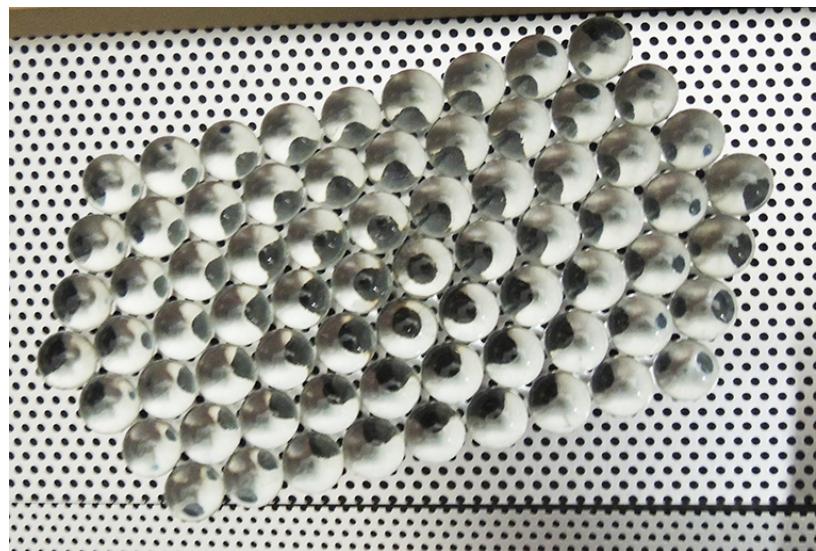


Fig. 4.6 穴にそって整列されたビー玉

さらに、この方法では視点と実験装置との距離によってひとまとめの黒点として認識される領域のサイズが大きく変化する。具体的には、視点が離れるほど黒い領域は大きくなり、視点が近づくほど黒い領域は小さくなる。4.3.1 節の結果から類推すると、ビー玉およびパンチングメタルをある角度前後から観測した場合、ビー玉の表面が黒色で占められると考えられ

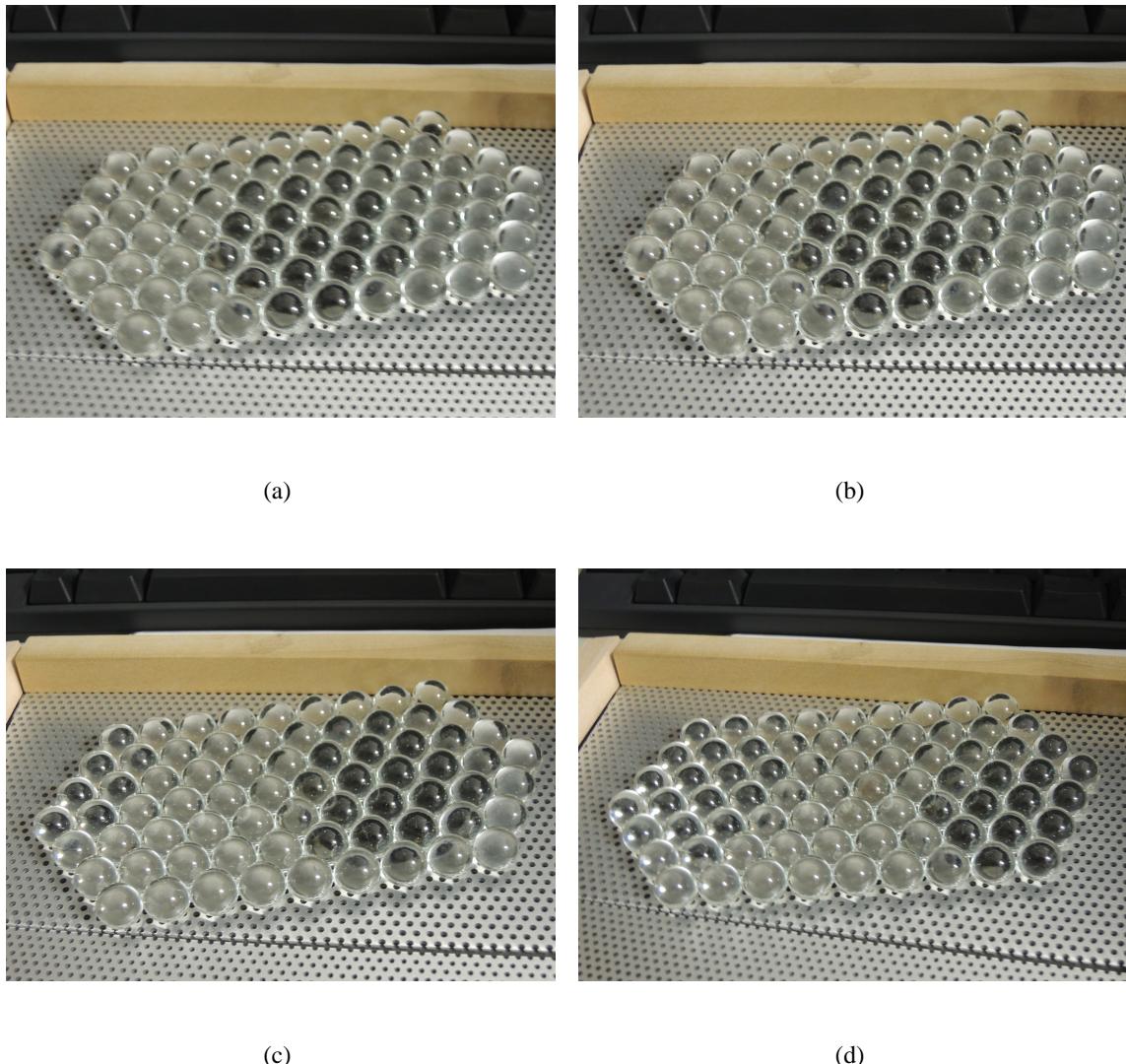


Fig. 4.7 視点を装置と平行に動かした場合

る。そして、視点が実験装置から離れるほど隣接する球と視線とのなす角度同士の差は小さくなり、視点が近づくほど隣接する球と視点とのなす角度同士の差は大きくなる (Fig.4.8)。すなわち、この実験装置上では視円錐のある角度に収まる領域は黒く見えると考えられる (Fig.4.9)。以上の性質は、実物の複眼上に現れる偽瞳孔現象を適切に説明することができ実際の偽瞳孔の観察結果とも合致するため、本研究において利用できる有力なモデルであると考えられる。

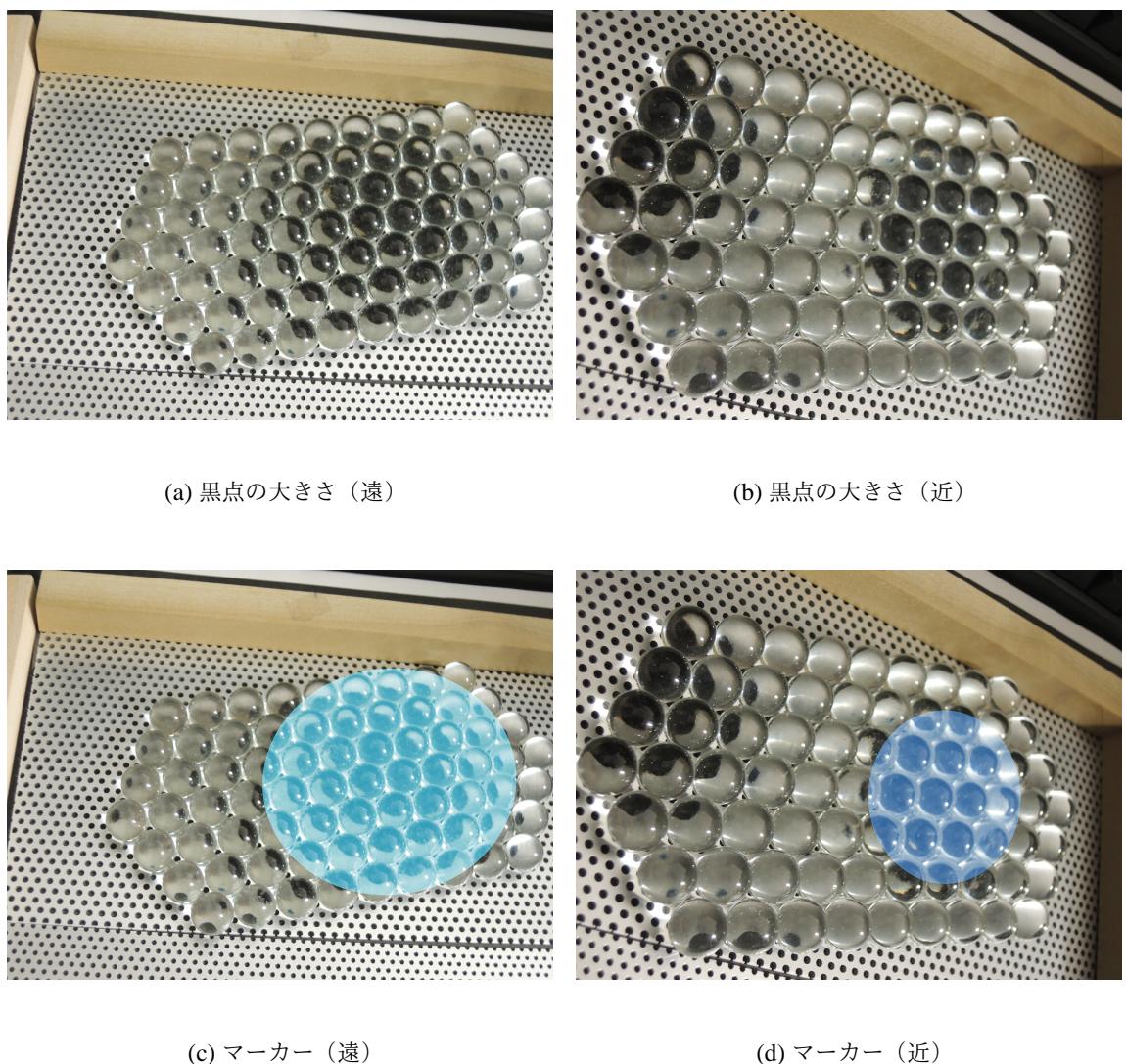


Fig. 4.8 黒点の大きさの違い

4.4 まとめ

複眼の性質を表現するための近似モデルを考えるため、永田の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。本実験の目的は3つあり、偽瞳孔のしくみを理解すること、実装前にモデルによる偽瞳孔の再現度を確認すること、そしてパラメータによる偽瞳孔の模様の変化を観察することであった。ビー玉、パンチングメタル、木の棒および画用紙を

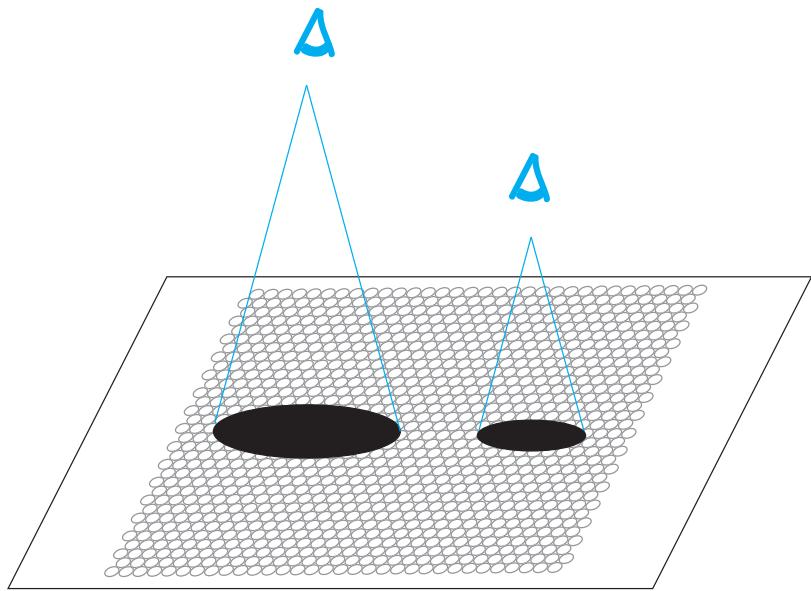


Fig. 4.9 視円錐と黒点の大きさ

用いて実験装置を作成し、いくつかのビー玉の配置パターンについてビー玉に写る像の観察を行った。4.3.1 節では、ビー玉の直下および周辺の黒点に対してビー玉が拡大レンズとして作用していることを確認した。4.3.2 節では、密集した球体レンズが奥で接している面を拡大できることを確認した。4.3.3 節では、レンズと黒点の位置を合わせることで拡大された黒点がよりはっきりと写ることと、その黒い領域が視点と実験装置との距離によって大きさを変えることを確認した。

偽瞳孔が生じるしくみは、個眼の凸レンズの拡大効果によって色素細胞の虚像が見えることによるものであると考えられる。レンズが密集して並ぶことでそれぞれの個眼が集合し、大きな偽瞳孔の黒点として現れる。個眼のレンズを球体であるビー玉で代替し、色素細胞を平面であるパンチングメタルに置き換えたモデルにおいても、偽瞳孔と同様の現象を確認した。個眼に相当するビー玉の大きさよりもおおきくはっきりとした黒色の領域を黒点の像として観測し、再現性としても実際の偽瞳孔と比較しても遜色のない結果を得ることができたと考えられる。球の大きさや黒点の配置などによる違いを確認することもでき、並んだ黒点同士の距離や表面と視点との距離が巨視的な偽瞳孔の模様に影響を与えることがわかった。本研究では、

4.3.3 節の場合を参考に複眼のモデル化を行い、アルゴリズムを考案する。

第 5 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、5.1 節および 5.2 節で本手法で用いた複眼の近似モデルを提案する。次に??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: Graphics Processing Unit）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 5.4 節以降では、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。

5.1 フレームワーク

5.1.1 複眼の光学モデル

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光学モデルを考案した (Fig.5.1)。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。テクスチャ情報は明暗のドットパターンから成り、その暗部は 3 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

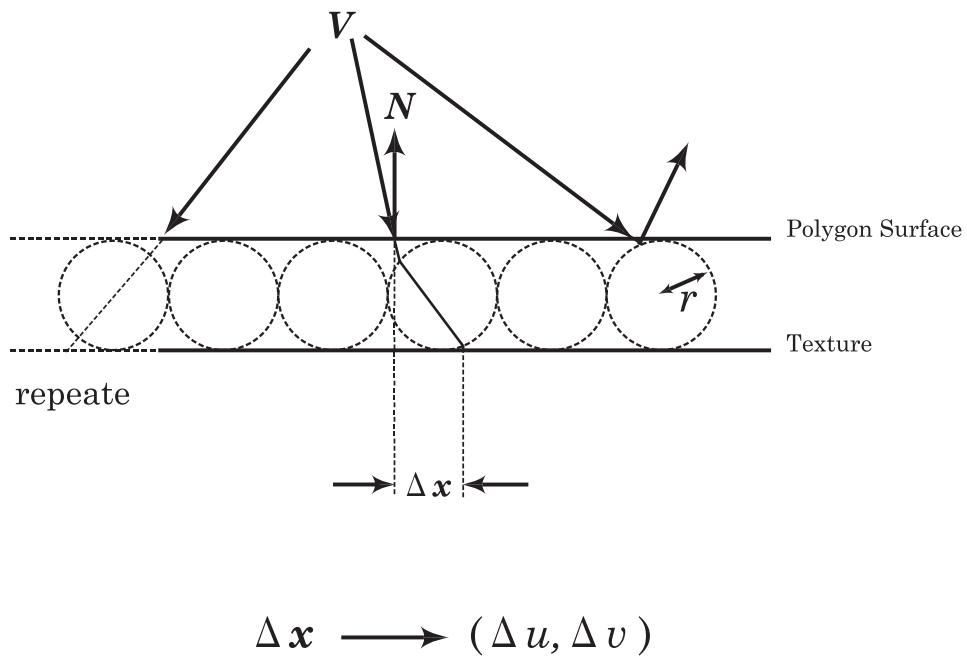


Fig. 5.1 シェーディングモデル

- 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。

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[5]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。

5.1.2 アルゴリズムの全体像

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.5.2)。アプ

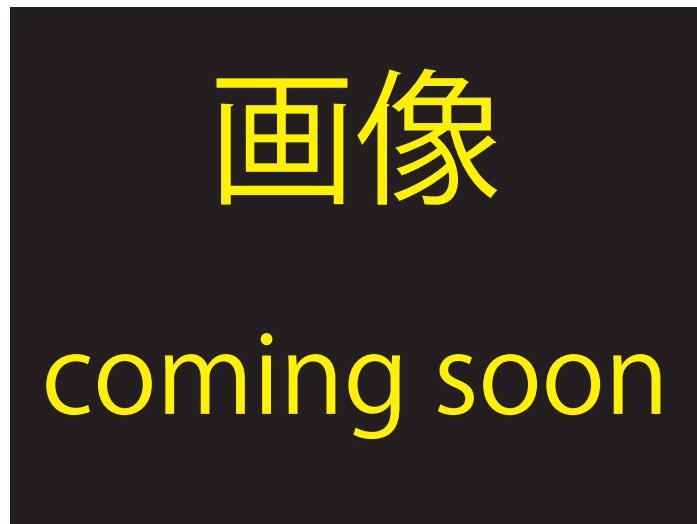


Fig. 5.2 アルゴリズムフレームワーク

リケーションでは入力する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 へ送る。複眼シェーダでは前述の光学モデルに基づき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。

5.2 テクスチャと球の配列

ひとつのテクスチャおよび球の組み合わせ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.5.3)。

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行う (Fig.5.4)。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、球の中心座標とポリゴンとの間には距離がある。球の半径を r として、第 4 章の実験装置に基づきポリゴンとテクスチャ平面との距離は $2r$ とした (Fig.??)。



Fig. 5.3 テクスチャと球の位置関係

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。すなわち、色の計算に用いるために用いるテクスチャと球の配置に利用するテクスチャは独立しており、計算上は別々の座標値を用いることが可能である。ここでは、両者を同一化しても計算上問題が無いため同じ座標値を用いた。

5.3 オブジェクトファイルデータから計算用データの生成

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。オブジェクトファイルとは、モデリングソフト等によりエクスポートされる形状データの集合体であり、一般にテキスト情報として処理することができる。オブジェクトファイル内に書き込まれる主なデータは以下のとおりである。

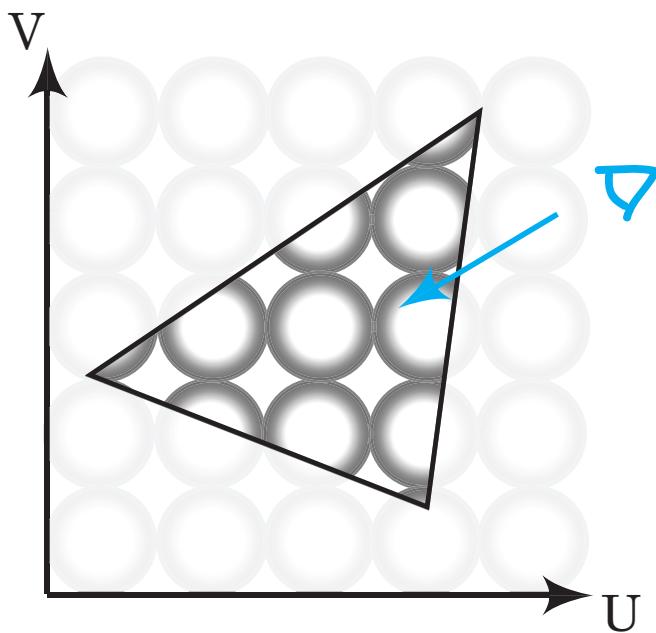


Fig. 5.4 テクスチャおよび球の格子状配置

- 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おり、この組み合わせが 3 つずつ続くデータを利用する。

計算用データの生成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。この処理は、オブジェクトファイルデータのテクスチャ座標系において距離 1 の範囲に何個個眼が配列しているかの情報である「個眼密度」を決定するためのものである。

$$\mathbf{T}_g = R_t \mathbf{T}_{obj} \quad (5.1)$$

ここで、 \mathbf{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、 \mathbf{T}_g は GPU 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とともに屈折レンズ（5.1.1 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。

5.3.1 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。

三角形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 $\mathbf{D}_u, \mathbf{D}_v$ として表すと、ポリゴン上の任意点 \mathbf{P} は $\mathbf{D}_u, \mathbf{D}_v$ を利用して以下のように表現することができる。

$$\mathbf{P} = \mathbf{P}_c + a\mathbf{D}_u + b\mathbf{D}_v \quad (5.2)$$

ここで、 a および b は点 \mathbf{P} におけるテクスチャ座標値 (a, b) を表している。 \mathbf{P}_c は既知の点 \mathbf{P}_e およびそのテクスチャ座標値 (a_e, b_e) によって

$$\mathbf{P}_c = \mathbf{P}_e - a_e\mathbf{D}_u - b_e\mathbf{D}_v \quad (5.3)$$

式 (5.3) のように表される。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 $\mathbf{D}_u, \mathbf{D}_v$ が既知であれば、式 (5.2) 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 $\mathbf{P}_c, \mathbf{D}_u$ および \mathbf{D}_v 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 \mathbf{D}_u および \mathbf{D}_v は以下の手順で求める。 \mathbf{D}_u および \mathbf{D}_v 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 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ とし、テクスチャ座標値を $\mathbf{T}_0, \mathbf{T}_1, \mathbf{T}_2$ 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 $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ および $\mathbf{T}_0, \mathbf{T}_1, \mathbf{T}_2$ から求めていく。

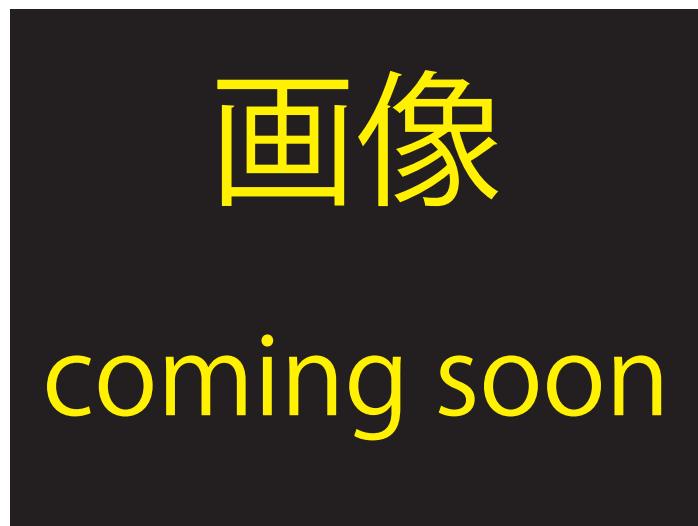


Fig. 5.5 ポリゴン上における頂点座標値およびテクスチャ座標値の関係

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

$$\mathbf{P}_{10} = \mathbf{P}_1 - \mathbf{P}_0 \quad (5.4)$$

$$\mathbf{P}_{20} = \mathbf{P}_2 - \mathbf{P}_0 \quad (5.5)$$

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

$$\mathbf{T}_{10} = \mathbf{T}_1 - \mathbf{T}_0 \quad (5.6)$$

$$\mathbf{T}_{20} = \mathbf{T}_2 - \mathbf{T}_0 \quad (5.7)$$

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.5.5)。

さて、テクスチャ座標軸方向 3 次元単位ベクトル \mathbf{D}_u および \mathbf{D}_v 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。

$$\mathbf{D}_u = i\mathbf{P}_{10} + j\mathbf{P}_{20} \quad (5.8)$$

$$\mathbf{D}_v = k\mathbf{P}_{10} + l\mathbf{P}_{20} \quad (5.9)$$

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



Fig. 5.6 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.5.6)。点 A の頂点座標値を \mathbf{P}_A そして点 B の頂点座標値を \mathbf{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。

$$\begin{aligned} \mathbf{P}_{AB} &= \mathbf{P}_A - \mathbf{P}_B \\ &= c\mathbf{P}_{10} + d\mathbf{P}_{20} \end{aligned} \quad (5.10)$$

さらに、 \mathbf{P}_{10} と \mathbf{T}_{10} および、 \mathbf{P}_{20} と \mathbf{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 \mathbf{T}_A そして点 B のテクスチャ座標値を \mathbf{T}_B とすると以下の式が成り立つ。

$$\begin{aligned} \mathbf{T}_{AB} &= \mathbf{T}_A - \mathbf{T}_B \\ &= c\mathbf{T}_{10} + d\mathbf{T}_{20} \end{aligned} \quad (5.11)$$

ここで、 i, j, k, l を用いて

$$i\mathbf{T}_{10} + j\mathbf{T}_{20} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (5.12)$$

$$k\mathbf{T}_{10} + l\mathbf{T}_{20} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (5.13)$$

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 (5.12) は U 軸単位ベクトル、式 (5.13) は V 軸単位ベクトルを表すことになる。さらに、式 (5.10) および式 (5.11) の対応関係から式 (5.8) および式 (5.9) を導くことができる。

2 次正方行列 $\mathbf{A} = (\mathbf{T}_{10}, \mathbf{T}_{20})$ とすると式 (5.12) と式 (5.13) から

$$\mathbf{A} \begin{pmatrix} i & k \\ j & l \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.14)$$

式 (5.14) が成り立ち、これを変形すると以下のようになる。

$$\begin{pmatrix} i & k \\ j & l \end{pmatrix} = \mathbf{A}^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \mathbf{A}^{-1} \quad (5.15)$$

以上から i, j, k, l を求めることができる。さらに、式 (5.3) において $\mathbf{P}_e = \mathbf{P}_0, (a_e, b_e) = \mathbf{T}_0$ として \mathbf{P}_c を作成し、 $\mathbf{D}_u, \mathbf{D}_v$ と合わせて利用する。

5.4 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 \mathbf{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 \mathbf{P} とし、その点のテクスチャ座標を $\mathbf{T} = (a, b)$ とすると 5.3.1 節の \mathbf{P}_c を用いて

$$\mathbf{P} = \mathbf{P}_c + a\mathbf{D}_u + b\mathbf{D}_v \quad (5.16)$$

と書ける。ここで、 a, b をそれぞれの値が正のとき

$$a' = \lfloor a \rfloor + 0.5 \quad (5.17)$$

$$b' = \lfloor b \rfloor + 0.5 \quad (5.18)$$

負のとき

$$a' = \lceil a \rceil - 0.5 \quad (5.19)$$

$$b' = \lceil b \rceil - 0.5 \quad (5.20)$$

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

$$\mathbf{P}' = \mathbf{P}_c + a' \mathbf{D}_u + b' \mathbf{D}_v \quad (5.21)$$

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

$$\mathbf{S} = \mathbf{P}' - r\mathbf{N} \quad (5.22)$$

とすると \mathbf{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 \mathbf{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 \mathbf{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 \mathbf{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

$$\mathbf{P}'_s = \mathbf{P}_s + h\mathbf{N} \quad (5.23)$$

とすればよい。 \mathbf{P}_s はポリゴン平面上の点なので

$$\mathbf{P}'_s = \mathbf{P}_c + a_s \mathbf{D}_u + b_s \mathbf{D}_v \quad (5.24)$$

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。

$$\mathbf{P}_{s'c} = \mathbf{P}'_s - \mathbf{P}_c \quad (5.25)$$

$$a_s = \frac{(\mathbf{D}_v \cdot \mathbf{D}_v)(\mathbf{P}_{s'c} \cdot \mathbf{D}_u) - (\mathbf{D}_v \cdot \mathbf{D}_u)(\mathbf{P}_{s'c} \cdot \mathbf{D}_v)}{(\mathbf{D}_u \cdot \mathbf{D}_u)(\mathbf{D}_v \cdot \mathbf{D}_v) - (\mathbf{D}_u \cdot \mathbf{D}_v)(\mathbf{D}_v \cdot \mathbf{D}_u)} \quad (5.26)$$

$$b_s = \frac{(\mathbf{D}_u \cdot \mathbf{D}_u)(\mathbf{P}_{s'c} \cdot \mathbf{D}_v) - (\mathbf{D}_u \cdot \mathbf{D}_v)(\mathbf{P}_{s'c} \cdot \mathbf{D}_u)}{(\mathbf{D}_u \cdot \mathbf{D}_u)(\mathbf{D}_v \cdot \mathbf{D}_v) - (\mathbf{D}_u \cdot \mathbf{D}_v)(\mathbf{D}_v \cdot \mathbf{D}_u)} \quad (5.27)$$

求めた a_s, b_s をもとに、式 (5.17) から式 (5.22) 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。交差していない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 5.5 節の処理へ進み、最後まで交差する球がなければ次の 5.6 節の処理へ進む。

5.5 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 \mathbf{C} と視線ベクトル \mathbf{V} および r から球と視線の交点 I_n が求まる (Fig.??)。点 I_n の座標値を \mathbf{I}_n とすると、これらから点 I_n における法線ベクトル \mathbf{N}_n は

$$\mathbf{N}_n = \frac{\mathbf{I}_n - \mathbf{C}}{\|\mathbf{I}_n - \mathbf{C}\|} \quad (5.28)$$

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 \mathbf{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。

5.5.1 隣接球処理

いちど球の内部へ入射し屈折した視線は再度、球から出射する際に屈折する。出射された視線が隣接する球へ入射する場合、さらに屈折を繰り返し、テクスチャ平面へ到達するまで球から球へと入射を繰り返す。ひとつの球に対する一連の屈折プロセスのうち、球外部からの視線が球と交差する点を I_n 、球内部を通過する屈折ベクトルを \mathbf{R}_{in} 、球と \mathbf{R}_{in} が再度交差する点を I_f 、そして球内部から球外部へと出射される屈折ベクトルを \mathbf{R}_{out} とする。

隣接する球へ到達する視線の挙動計算を隣接球処理と呼ぶことにする。本手法では、球体レンズを格子状に配置しているため、ひとつの球に隣接する球は斜め方向も含めると 8 つあると考える。しかし、一度の隣接球処理のたびに 8 つの球それぞれに対して視線が通過するかどうかを判定するのは冗長である。そこで、本手法ではポリゴン平面上の空間を 4 つの領域に分け (Fig.5.8)、視線がどの領域へ進むかをベクトルの評価によって判別する。これによって、一度の隣接球処理で最大 3 つの球に対して判定を行うだけで済むようになる。

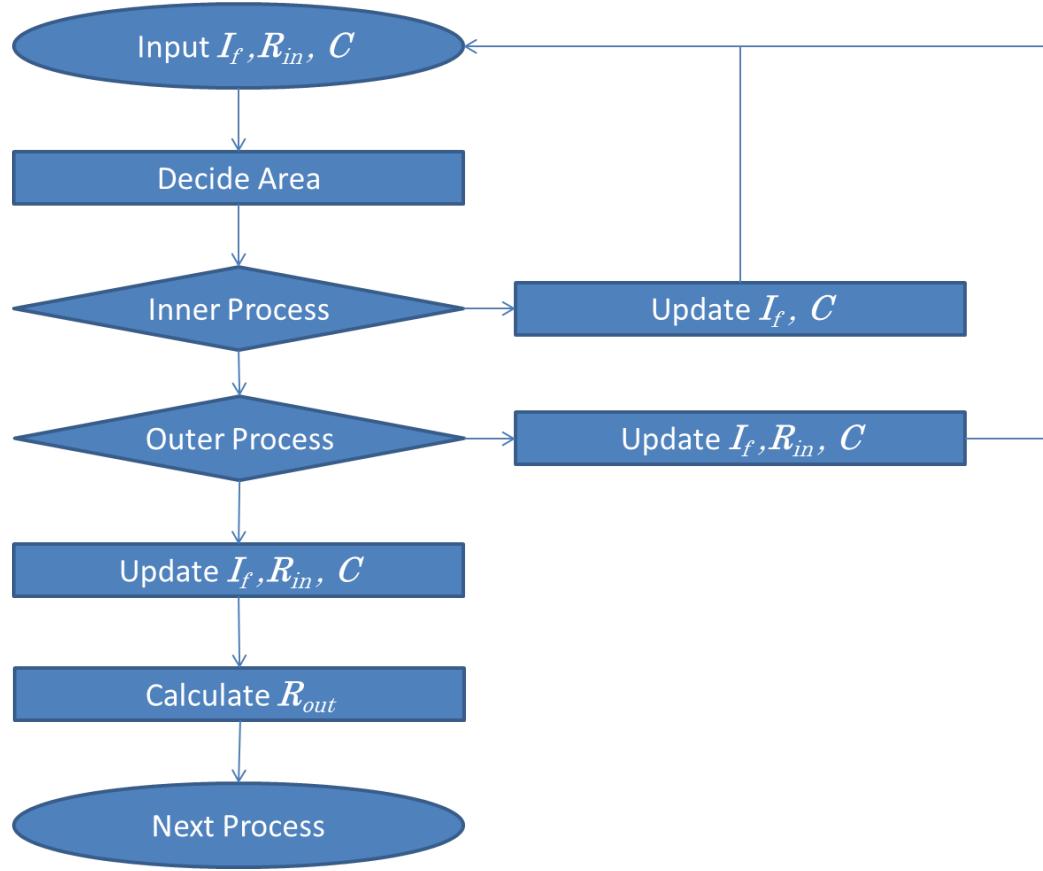


Fig. 5.7 隣接球処理アルゴリズム

視線方向は3次元ベクトルであるため、平面上の方向ベクトル \mathbf{R}_{dir} を求めるためには視線方向ベクトルを平面上へ投影する必要がある。この投影処理は、正規化された球内部を通過する屈折ベクトル \mathbf{R}_{in} および法線ベクトル \mathbf{N} を用いて以下のように表される

$$\mathbf{R}_{dir} = \mathbf{R}_{in} - (\mathbf{R}_{in} \cdot \mathbf{N})\mathbf{N} \quad (5.29)$$

さらに、 $\mathbf{R}_{dir} \cdot \mathbf{D}_u$ と $\mathbf{R}_{dir} \cdot \mathbf{D}_v$ の絶対値の大小比較およびそれぞれの符号によって視線ベクトル（ここでは屈折ベクトル \mathbf{R}_{in} ）の進行する領域を決定することができる ((Fig.5.7) の Decide Area プロセス)。領域決定後は、それぞれの領域に属する隣接球に対し順次処理を行っていく。続く、(Fig.5.7) の Inner Process では、視線が球の内部から隣接球の内部へと通過するかどうかの判別を行っている。これは、球同士の中心距離よりも球の直径が大きい場合に、球同士

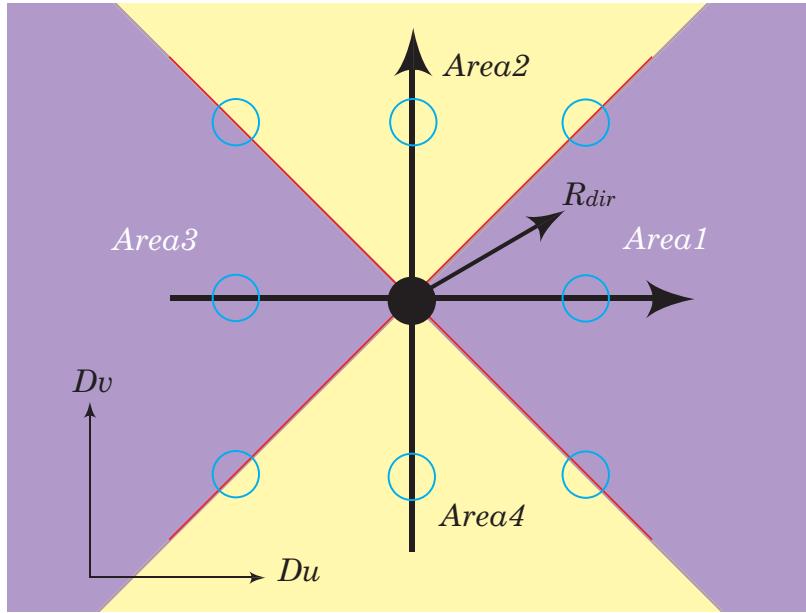


Fig. 5.8 領域分割

がオーバーラップするため必要となる処理である。この処理では、まず球と R_{in} の交点 I_f を求め、 I_f が隣接球 C_n の内部にあるかどうかを判定し、内部にある場合は C を C_n として更新し、新たな C に対して再度 I_f を計算する。すなわち、 C および I_f を更新し隣接球処理プロセスの先頭へ戻る。領域内のすべての隣接球の内部にはじめの I_f がない場合、続く Outer Process に進む。

Outer Process ではいちど球から視線が出射されたあと隣接球に再度入射する場合の計算を行う。Inner Process と異なるのは、視線の光が進む媒質が球内部から球外部へと変化するため、余計に屈折が起こる点である。視線が領域内の隣接球のうちどれかに再度入射する場合、隣接球と視線との交点計算によって R_{in}, I_f を求め、 C を C_n として更新する。すなわち、 C および R_{in}, I_f を更新し隣接球処理プロセスの先頭へ戻る。

Inner Process および Outer Process でどの隣接球とも交差しないか、もしくは設定したループ処理の上限値に達した場合、 C, R_{in}, I_f を更新し、それらから R_{out} を算出して次の処理へと進む。

5.6 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（5.1.1 節）へと到達する。テクスチャ平面上の到達点を \mathbf{P}_T とすると、この点におけるテクスチャ座標値は 5.4 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。

$$\mathbf{P}'_T = \mathbf{P}_T + 2r\mathbf{N} \quad (5.30)$$

座標 \mathbf{P}'_T はポリゴン平面上にあるため式 (5.24) と同様に表すことができる。すなわち、求める座標は式 (5.26) および式 (5.27) 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。

5.7 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 \mathbf{c}_t とすると、最終的に描画される色 \mathbf{c} は次の用に計算される。

$$\mathbf{c} = \mathbf{c}_t(\mathbf{a} + \mathbf{d}) + \mathbf{s} \quad (5.31)$$

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

第 6 章

結果と考察

6.1 実装方法および実行環境

本手法ではグラフィックスライブラリとして OpenGL(オープンジーエル: Open Graphics Library) を利用し、プログラムの実装にあたっては C++ 言語を用いている。GPU 上で動作するプログラムの実装にあたっては、GLSL(ジーエルエスエル: OpenGL Shading Language) を用いた。本手法を適用する形状データは Autodesk Maya 2013 で作成したオブジェクトファイルを使って作成した。

プログラムの実行に使用した計算機のハードウェア構成は (Table 6.1) のとおりである。

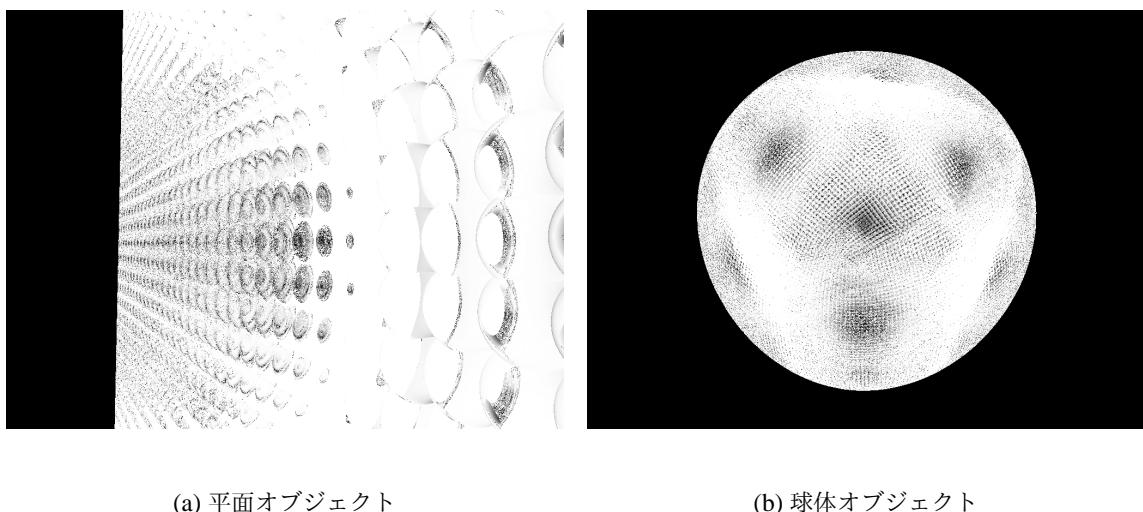
Table 6.1 実行環境

CPU	Intel Core i7 960 3.20 GHz
メモリ	30 GB
GPU	NVIDIA GeForce GTX 660

6.2 処理系

6.2.1 内部通過処理および外部通過処理

第5章で述べた隣接球を通過する光の挙動について、平面オブジェクトおよび球体オブジェクトを例に問題点を指摘する。本手法では、球体レンズ同士の間にできる隙間を埋めるために、球の直径を球同士の中心距離よりも大きく設定している（5.5.1節）。また、複数の球を通過する光を想定し、光が球の外部を経由して隣接する球に入射する場合のアルゴリズムを実装していた。しかし、外部を経由する場合の実装（以下、外部通過処理）によって得られる結果画像にはノイズが多く、求める画像とは異なる結果となってしまった（Fig.6.1）。そのため、手



(a) 平面オブジェクト

(b) 球体オブジェクト

Fig. 6.1 外部通過処理に見られるノイズ

法の修正として外部通過処理を実装から排除し、球の内部から内部への光の通過を考慮した実装（以下、内部通過処理）のみを採択することにした。

外部通過処理を排除しても、本手法で想定される範囲内では問題はないと考えられる。理由として、外部通過処理と内部通過処理のそれぞれによって得られる画像にはノイズの有無以外

に大きな差異が観測されなかつたことと、球の半径を十分に大きい値で設定すれば、球の外部を経由して隣接する球に入射する光はあまり存在しないと推測されることが挙げられる。ゆえに、本章での考察は主に内部通過処理によって得られるものに対して行う。

6.3 ジオメトリ

6.3.1 レンズ半径と偽瞳孔の数

本手法では、レンズの位置座標とレンズの半径は独立して扱うことができる。6.2.1 節でも述べたように、本手法では配置された球体レンズ同士の間の隙間を埋めるため、球の半径を球同士が接するよりも大きい値で設定しており、球の半径によって光の入射する領域のレンズの曲率が変化する。すなわち、球の半径 r の設定値によって複眼の外観が変化するため、両者の関係について考察を行う。以下ではテクスチャに後述の (Fig.6.4(a)) を使用しており、テクスチャ解像度 $R_t = 300$ を使用している。

比較画像 (Fig.6.2) からレンズ半径が大きくなるにつれて観測される偽瞳孔の数は増加していることがわかる。これは、半径が小さく曲率が大きければ個眼の端でのレンズ表面への進入角が小さくなるので、より内側への屈折が起こり、逆に半径が大きく曲率が小さければ、進入角が相対的に大きくなるので屈折の度合いが抑えられ、結果的に隣接する色素細胞テクスチャへ到達する頻度が高くなるためであると考えられる。

再現性を高めるためには、対象生物の個眼のレンズの曲率を測定するか、ユーザが任意の半径値 r を設定することが必要となるが、ある程度恣意的に結果を操作できる点はユーザビリティにつながるのではないかだろうか。事実、生物種によって観察される偽瞳孔の密度や分布は異なるため、求める結果を得るために試行錯誤が必要である。

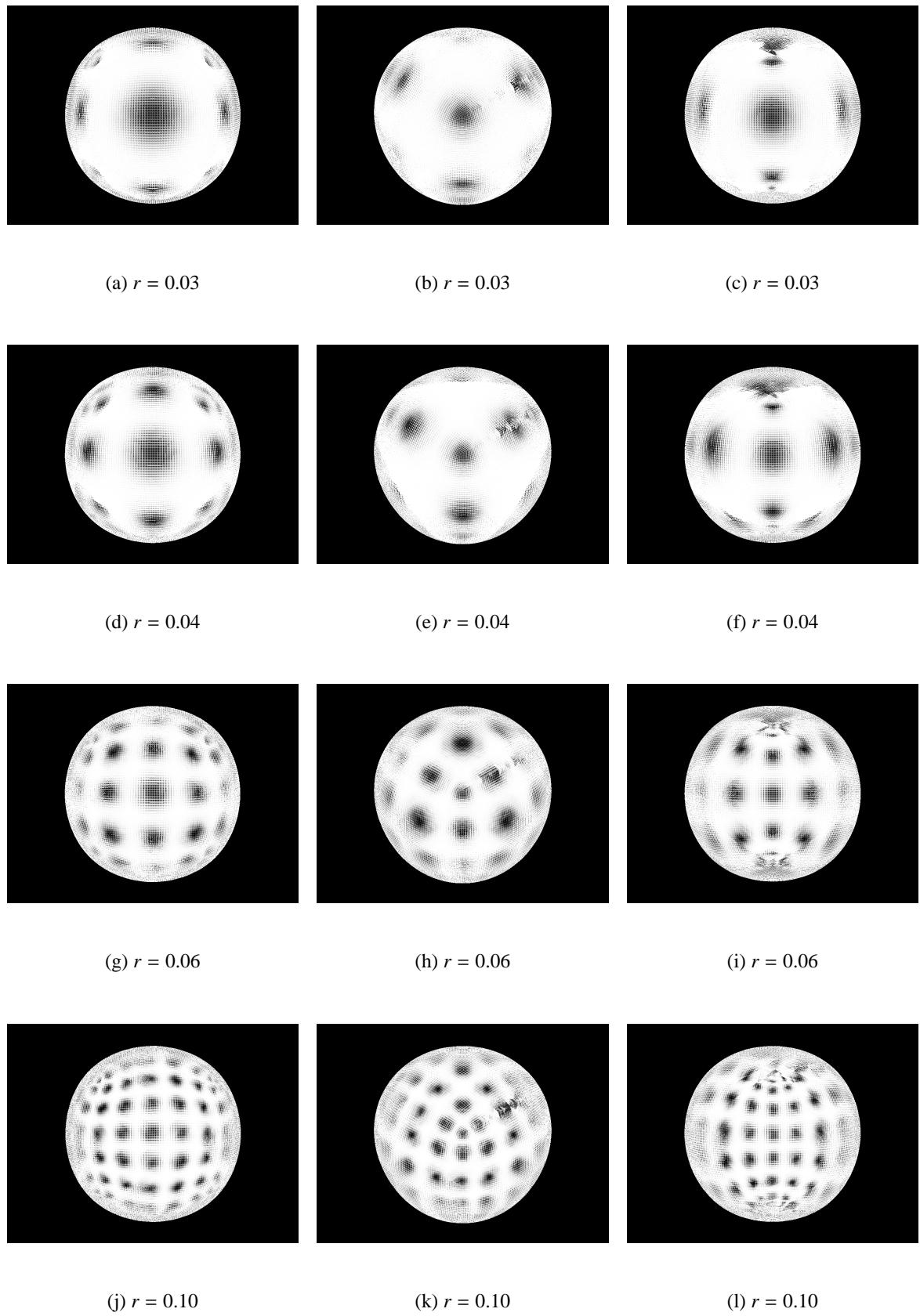
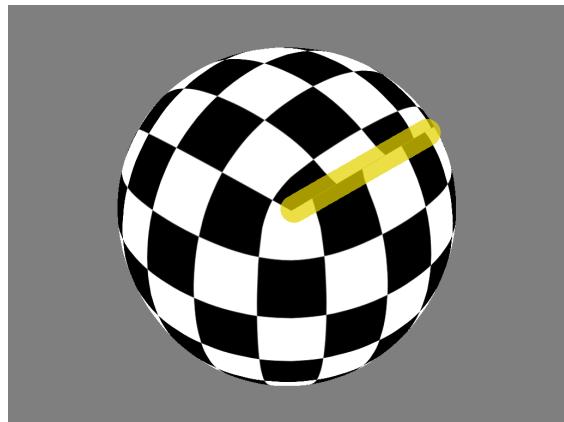


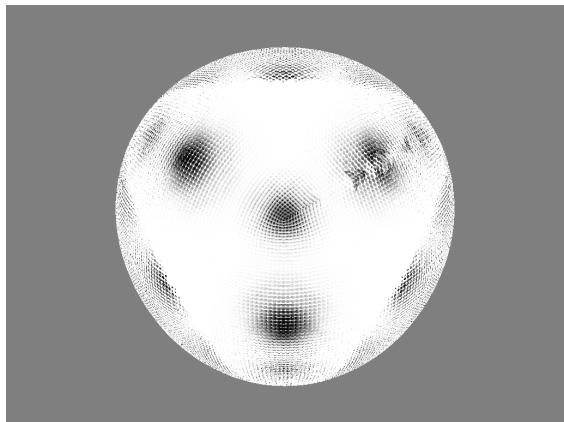
Fig. 6.2 レンズ半径 r と偽瞳孔の数の関係

6.3.2 歪みとテクスチャの関係

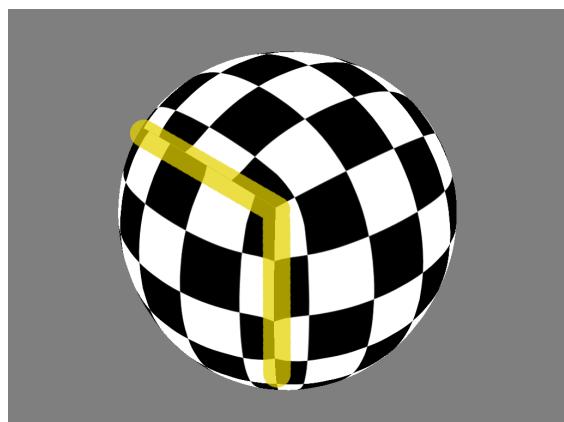
結果画像の一部には歪みが生じている部分が存在する (Fig.6.3)。これは、オブジェクトに適用されたテクスチャに継ぎ目が存在し、なめらかに接続していないことが原因である。本手法の特性上、曲面オブジェクトの表面に対しては格子状にテクスチャを適用する必要があるため、立体表面でテクスチャの歪みが生じることは避けられない。すなわち、閉じた立体ではなく開いた曲面もしくはテクスチャの継ぎ目を表示しないモデルを使用するなどの工夫が要求される。また、この問題はレンズ球の配置条件によって生じるので、レンズ球の配置アルゴリズムを改善するなどの対応策が今後求められる。



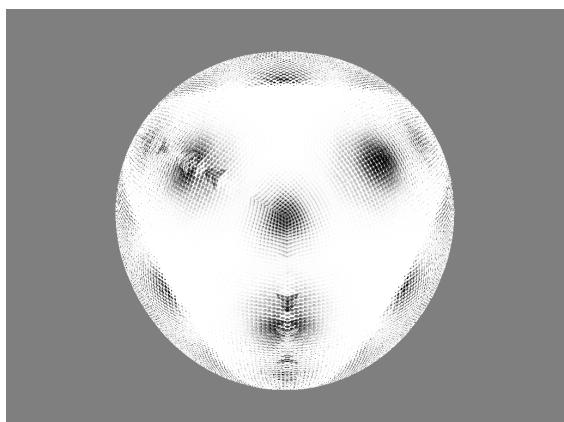
(a) テクスチャマップ



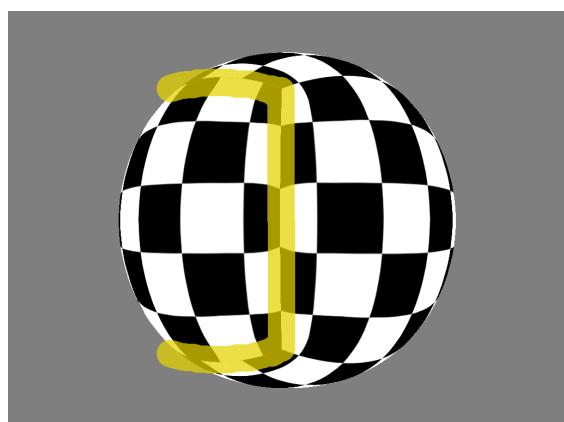
(b) 結果



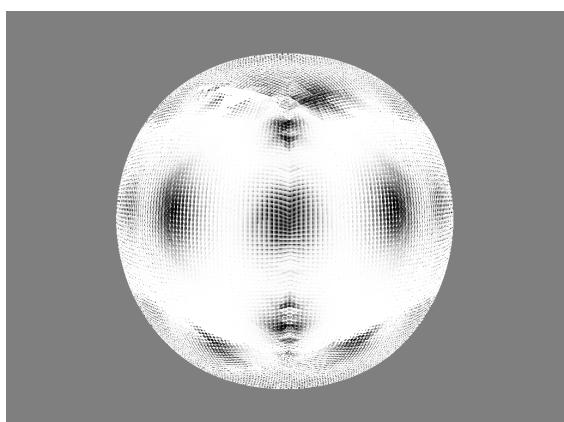
(c) テクスチャマップ



(d) 結果



(e) テクスチャマップ



(f) 結果

Fig. 6.3 テクスチャの継ぎ目

6.3.3 さまざまなテクスチャ

本手法ではテクスチャを変更することにより外観が大幅に変化する。基本的には、モデルであるパンチングメタルを模して黒点のテクスチャをタイリングする。以下に、本手法で用いた基本のテクスチャおよび任意のテクスチャを使用した場合の結果を紹介する (Fig.6.4)。

基本のテクスチャは球の中心と黒点の中心を合わせる必要があるため、画像中心から同心円上に広がる黒点を使用している。同じ黒点のテクスチャ (Fig.6.4(a)), (Fig.6.5(e)) であっても (Fig.6.4(b)) と (Fig.6.5(f)) のように、得られる結果画像は大きく異なることがわかる。また、(Fig.6.5(a)), (Fig.6.5(b)) および (Fig.6.5(c)), (Fig.6.5(d)) からは、敷き詰められた球体レンズによって基本となるテクスチャの画像が拡大されていることが確認できる。よって、本手法は4.3 節の結果を反映していると言える。

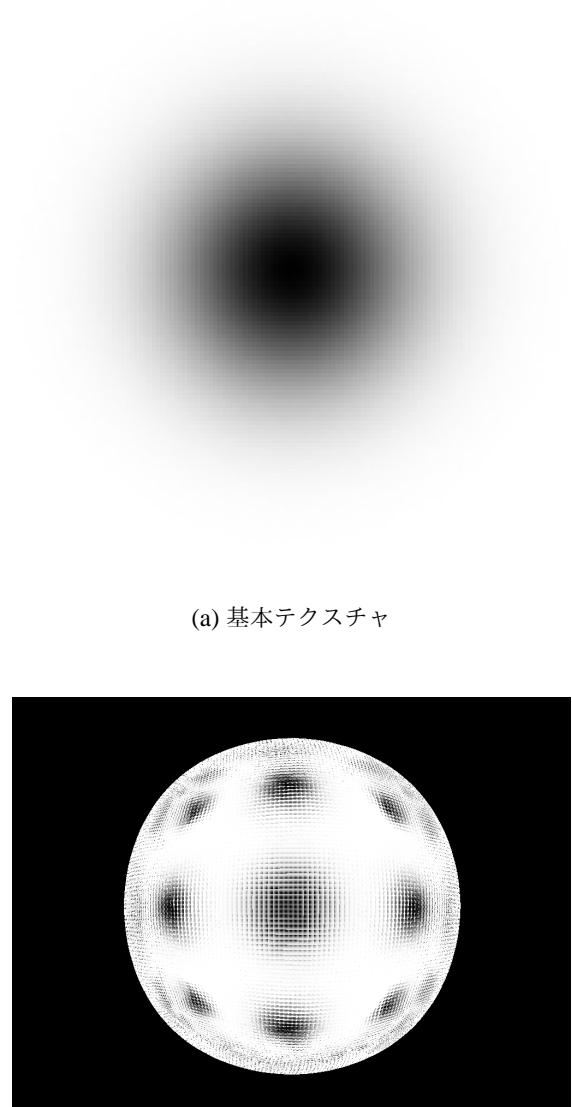


Fig. 6.4 基本テクスチャと適用結果

6.4 パラメータと実行速度

それぞれのパラメータに応じて得られる結果画像や実行速度について比較、検討を行う。評価を行うパラメータは光が通過する球の数の上限値、視点との距離、個眼密度である。カメラ

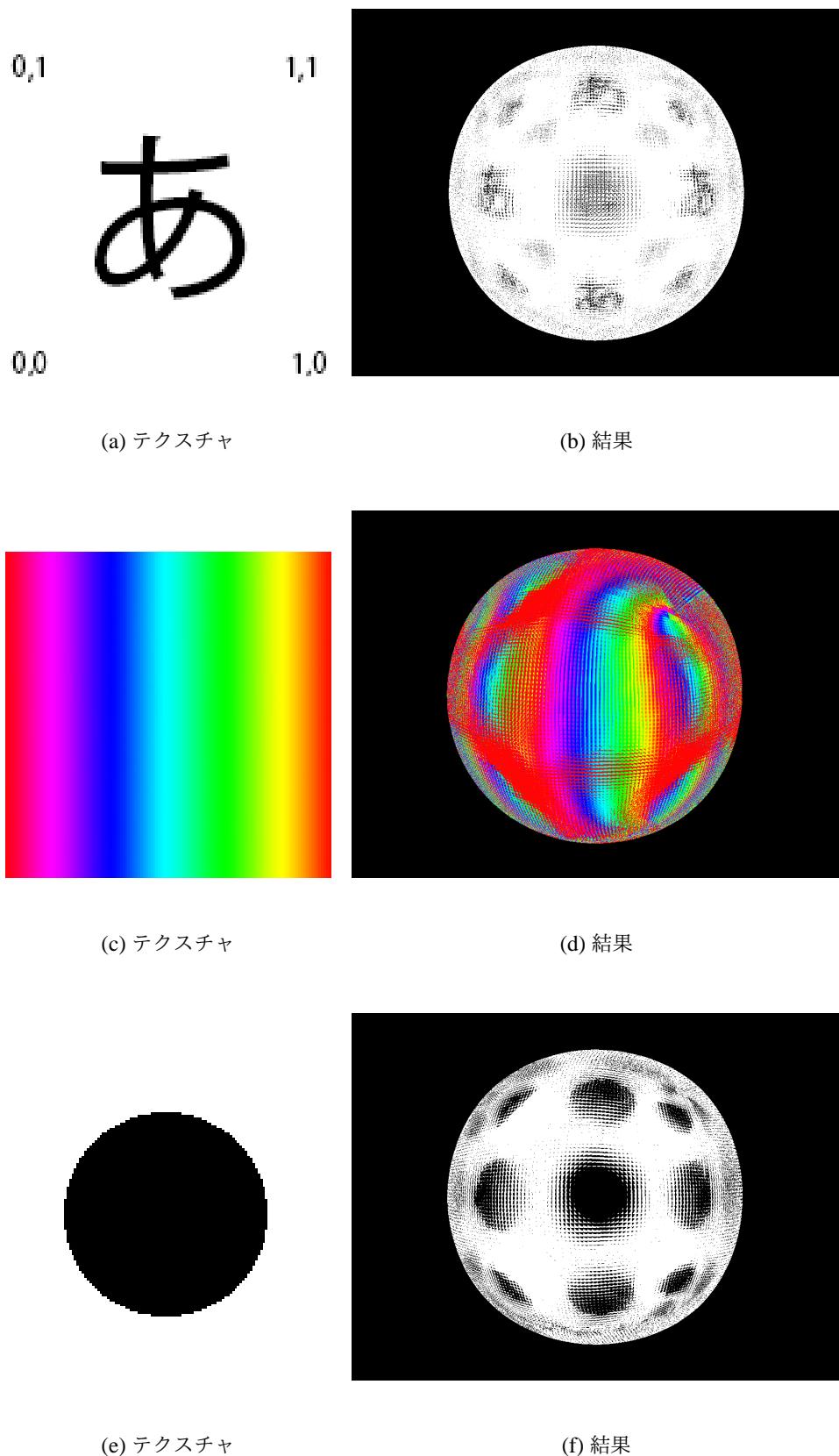
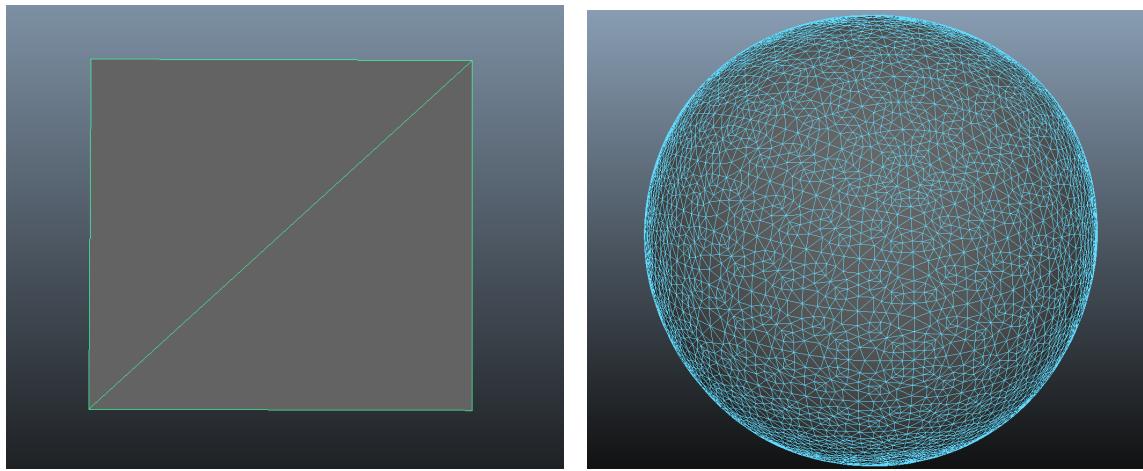


Fig. 6.5 テクスチャの変更による外観の変化

の垂直画角およびレンズの屈折率は固定する。第4章でレンズとして用いたビー玉はガラス製であるため、屈折率はガラスの文献値 1.491 を用いる。対象オブジェクトは以下に示す平面オブジェクトおよび球体オブジェクトとする。



(a) 平面オブジェクト (ワイヤフレーム)

(b) 球体オブジェクト (ワイヤフレーム)

Fig. 6.6 評価オブジェクト (ワイヤフレーム表示)

Table 6.2 オブジェクト情報

	平面オブジェクト	球体オブジェクト
頂点数	4	9218
面の数	2	18432
ポリゴンの種類	三角形	三角形

6.4.1 隣接球の処理 ON/OFF

光がどの程度の回数、隣接球を通過する必要があるのかについての考察を行う。光が複眼表面に対して平行に近い角度で入射した場合、光はひとつの球だけではなくふたつ以上の球を通過すると考えられる。通過する球の数が増えればそれだけ処理に時間がかかるため、通過する球の数には上限値を設定する必要がある。すなわち、物理的な正確さと処理速度はトレードオフ

フの関係にあると言える。

では、通過する球の数の上限値を変化させることによって得られる結果について議論を行う。上限値を設定することによる操作をここでは隣接球処理と呼ぶことにする。以下では上限値 $l = 0, 1, 2, 3, \dots$ のように設定し、上限値 l が 0 の場合には隣接球処理を行わないこととする。すなわち、 $l = 0$ の場合、最初の球に入射した光は他のいかなる隣接球にも入射せずテクスチャ平面(5.1.1 節)へ到達するものとして考える。以下にまず、各 l の値に対する結果を示す(Fig.6.7)(Fig.6.8)。テクスチャは (Fig.6.4(a)) を使用しており、球の半径 r の値は約 3.3×10^{-2} 、テクスチャ解像度 $R_t = 300$ を使用している。

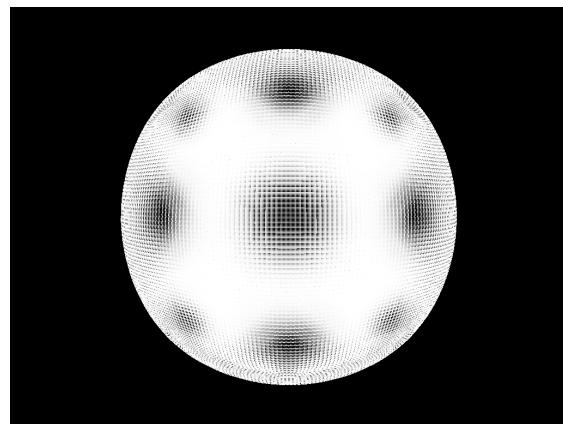
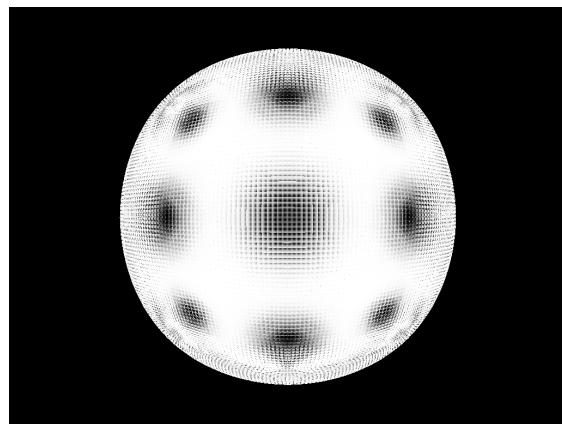
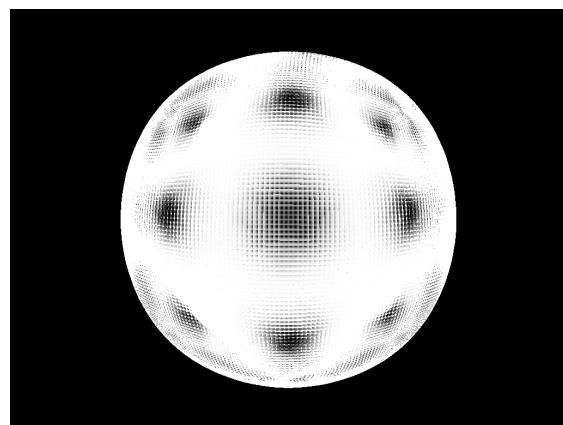
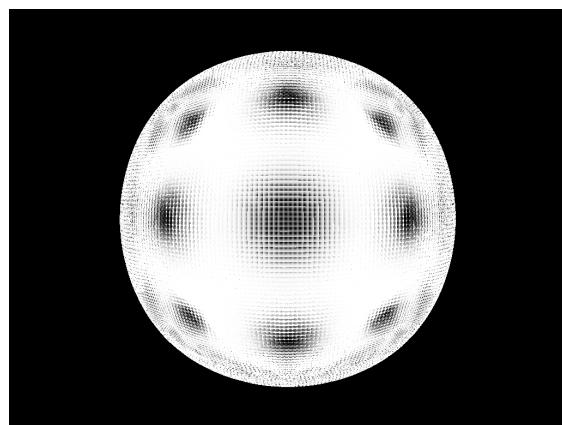
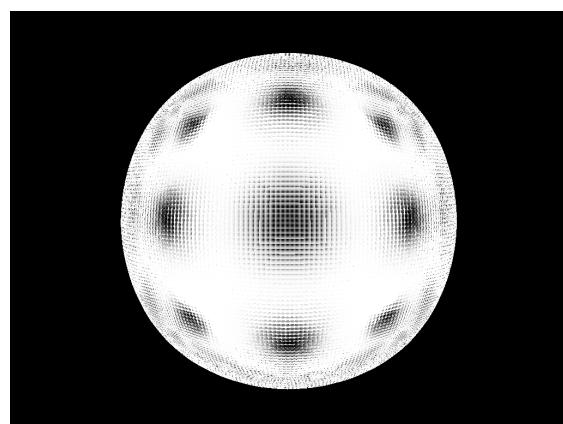
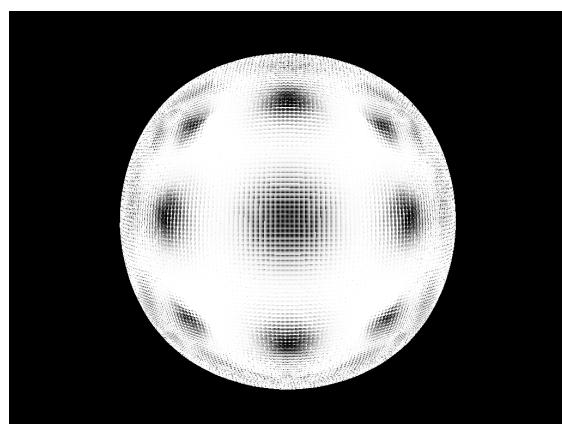
(a) $l = 0$ (b) $l = 1$ (c) $l = 2$ (d) $l = 3$ (e) $l = 5$ (f) $l = 10$

Fig. 6.7 上限値による画像の変化 1

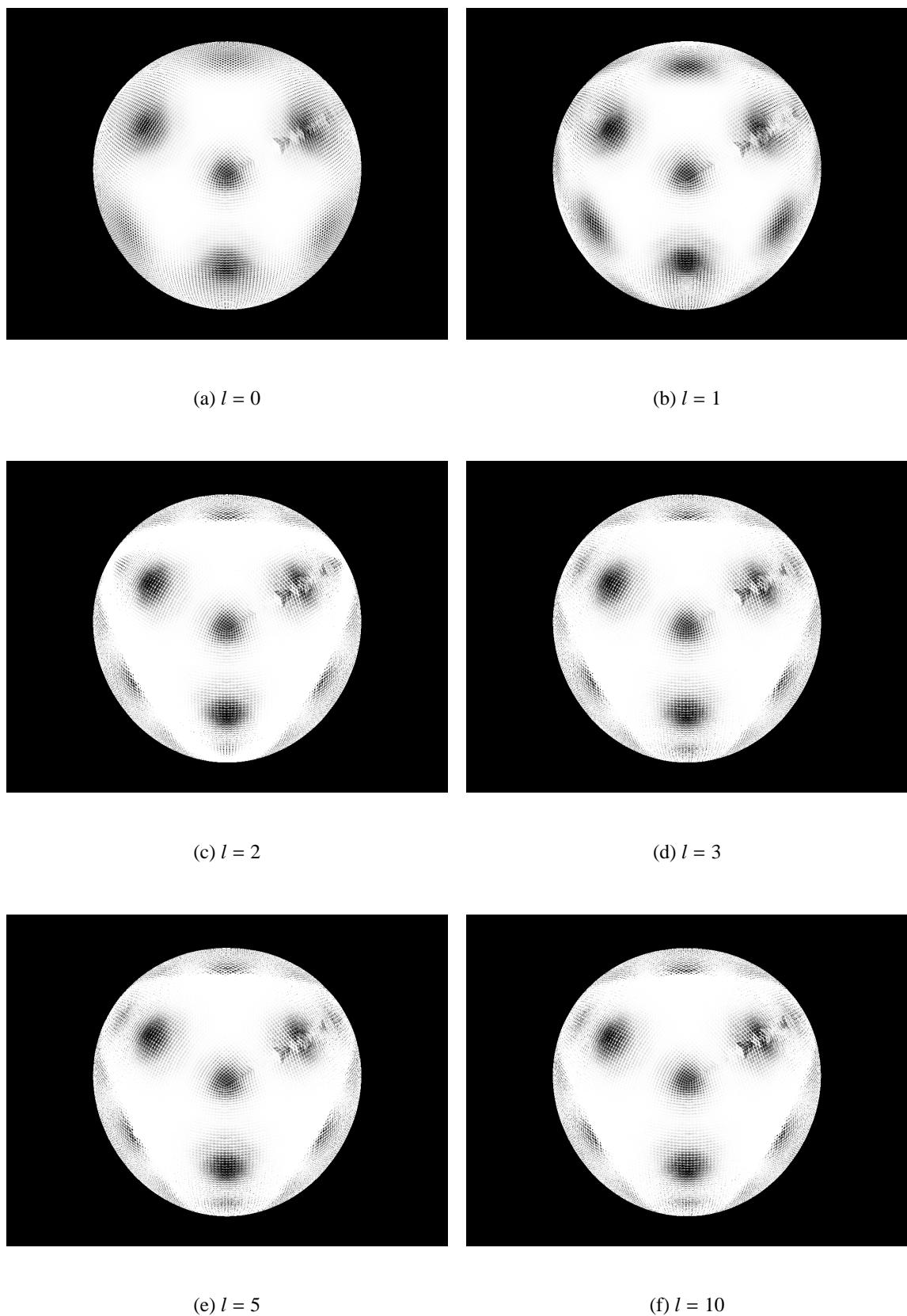


Fig. 6.8 上限値による画像の変化 2

結果画像からは $l = 0$ の場合とそれ以外とで顕著な差異が見られ、 $l = 1$ および $l = 2$ の場合とそれ以外の場合でわずかな差異が確認できた。また、 $l = 3$ 以上の場合については肉眼で判別できるほどの差異は見つからなかった。まず、 $l = 0$ の場合とそれ以外との差異について述べる。 $l = 0$ の場合、side pupil(3.3.2 節) の存在が確認できるものの、他の場合と比較するとやや黒色が薄く、隣接球処理をした場合のほうがはっきりとした side pupil となっていることがわかる。 $l = 1, 2$ の場合とそれ以外の場合を代表して $l = 3$ の場合について比較を行うと、球体オブジェクトの縁において有意差が見られる。 $l = 1$ の場合、side pupil のさらに外側に位置する second side pupil((Fig.6.11)(Fig.6.12) 緑部) のあたりは $l = 3$ の場合よりも黒点の形状が曖昧となっている。逆に、 $l = 2$ の場合は $l = 1, 3$ の場合と比較して白色領域が広くなっている。特に、 $l = 1$ の場合では (Fig.6.11)(Fig.6.12) 黄部付近の second side pupil がやや大きいのに対して、それ以降の場合では縮小していることが顕著に観測される。以上のことから、偽瞳孔の side pupil および second side pupil には隣接球を通過する光が影響していることがわかり、3.3.2 節の理論と一致する。

続いて、実行速度と l の関係を示す (Fig.6.10)。平均値 (Fig.6.10(a)) と実測値 (Fig.6.10(b)) のいずれのグラフからも、実行速度と通過隣接球上限値 l は線形の関係であることがわかる。さらに、リアルタイムの基準が 60 fps であるとすると、今回の球体オブジェクトの場合では $l = 900$ 付近がリアルタイムの限界値であると考えられる。また、先述のように $l = 3$ 以上では結果画像に有意な差が見られないことから、 l の設定値は 10^1 オーダーで十分であると言える。

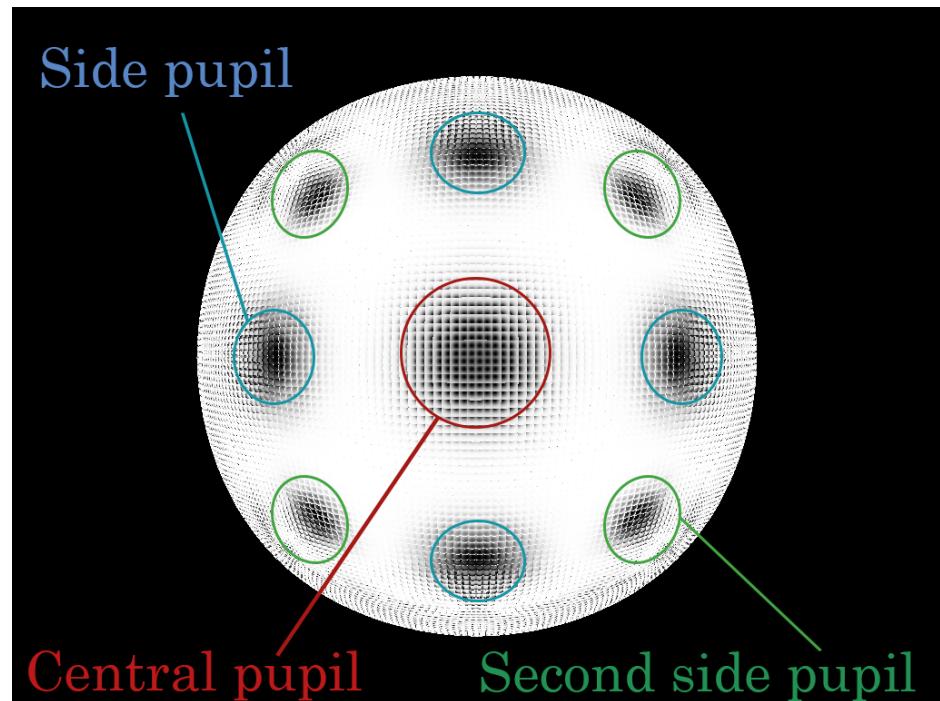
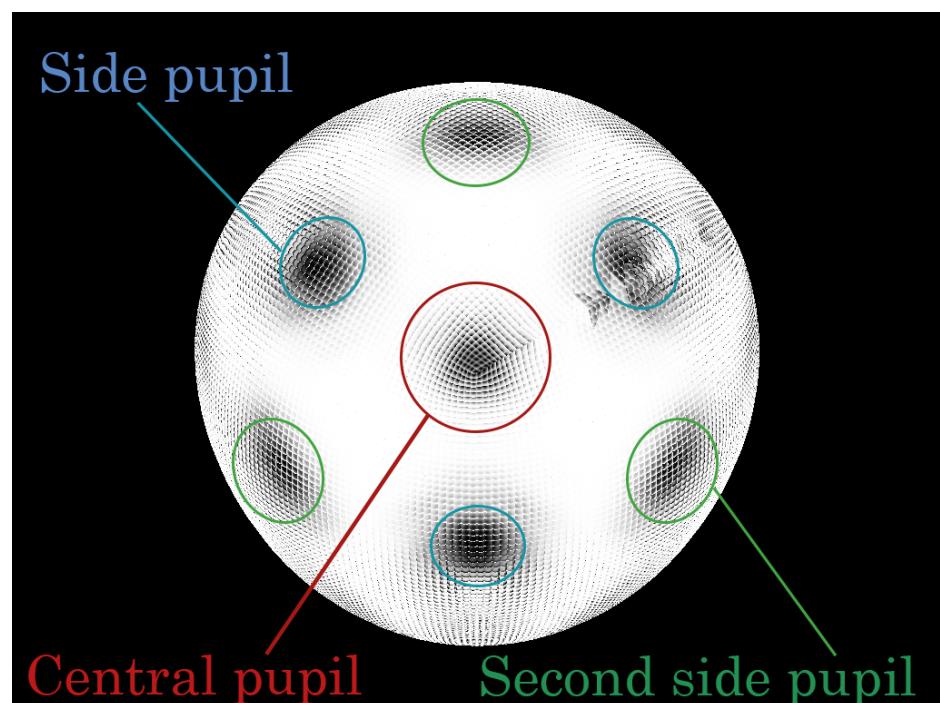
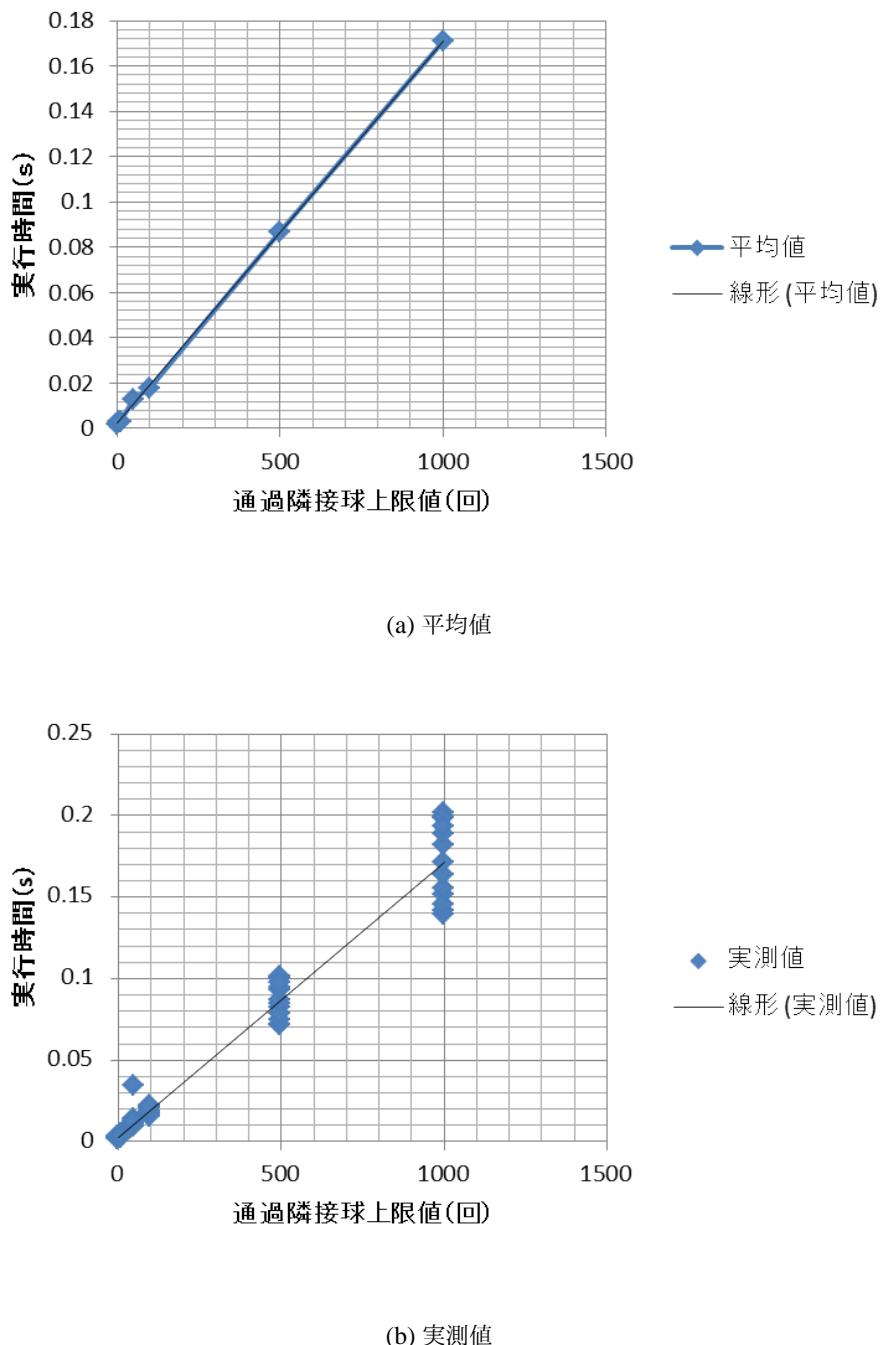
(a) $l = 1$ (b) $l = 1$

Fig. 6.9 各偽瞳孔の名称および場所

Fig. 6.10 実行速度と l の関係

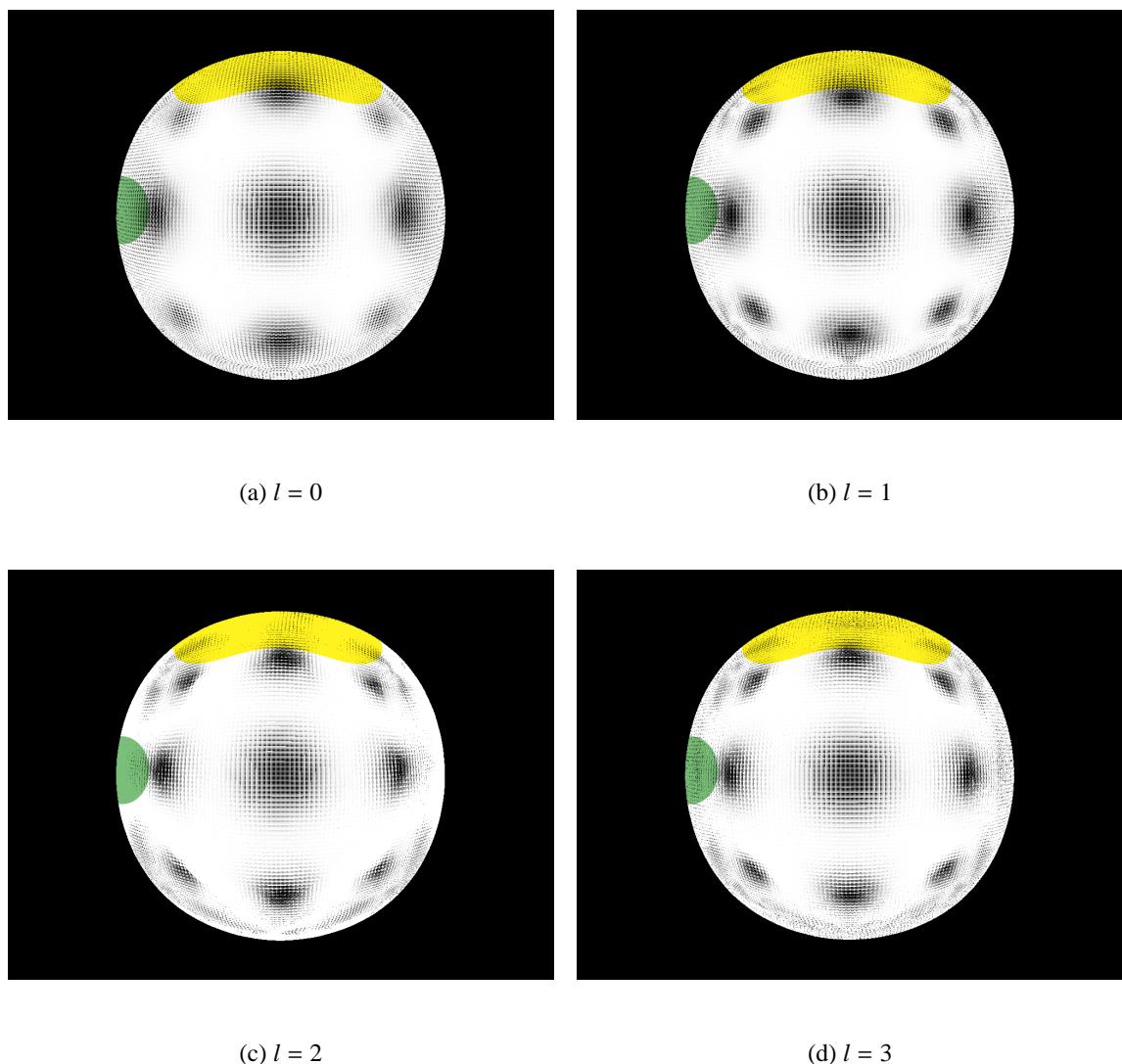


Fig. 6.11 上限値による画像の変化比較 3

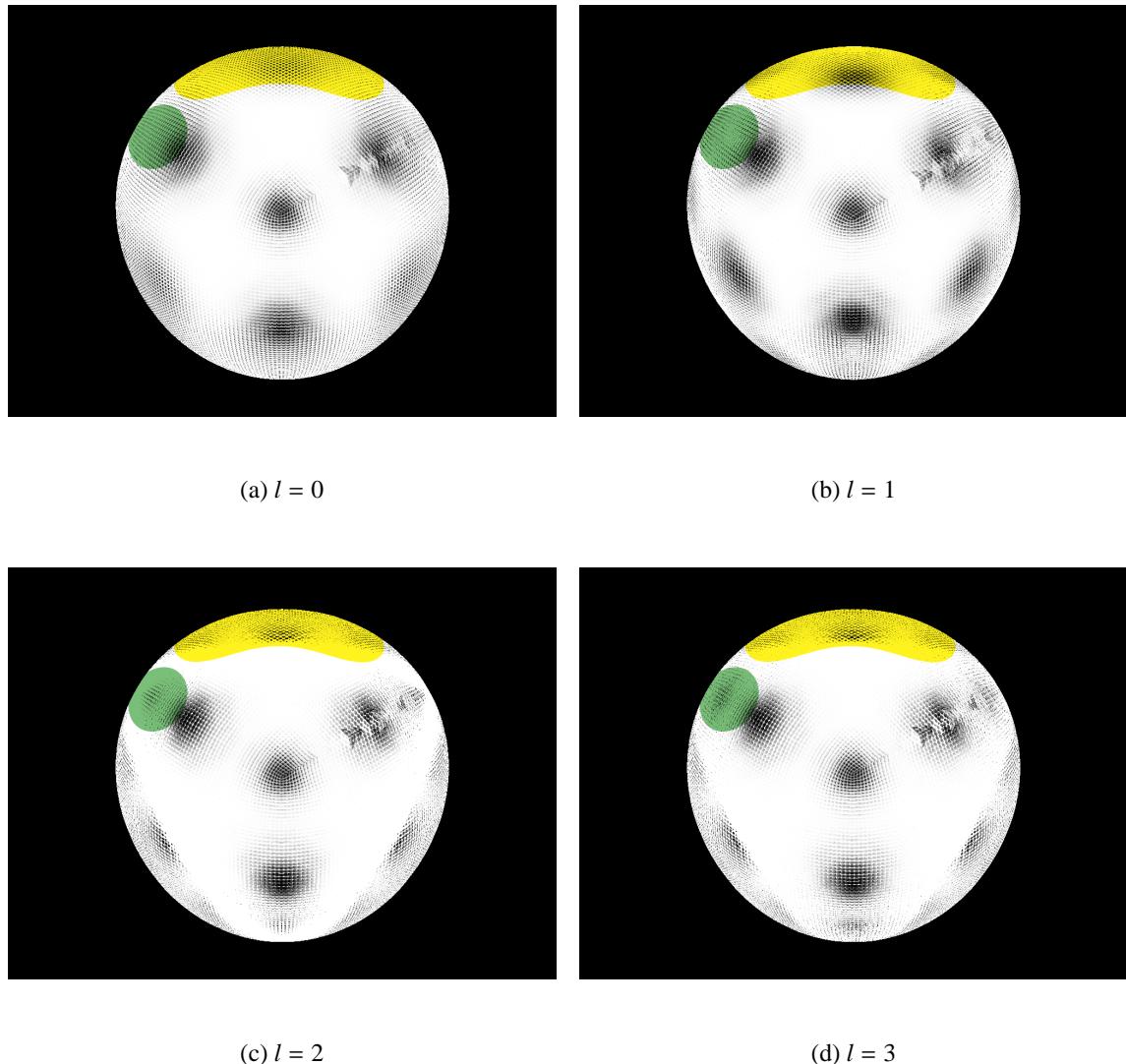


Fig. 6.12 上限値による画像の変化比較 4

6.4.2 視点との距離による変化

4.3.3 節では、偽瞳孔に相当する黒点の、装置に対する大きさは視点との距離によって変化した。ここでは、視点とオブジェクトの距離を変化させた場合の本手法の実行結果について論じる。第4章の結果では、視点が近づくほど黒点の大きさは相対的に小さくなり、黒点同士の距離は短くなった。(Fig.6.14) および (Fig.6.13) は隣接球処理の上限値および個眼密度を固定して視点との距離を変化させた場合の実行結果である。

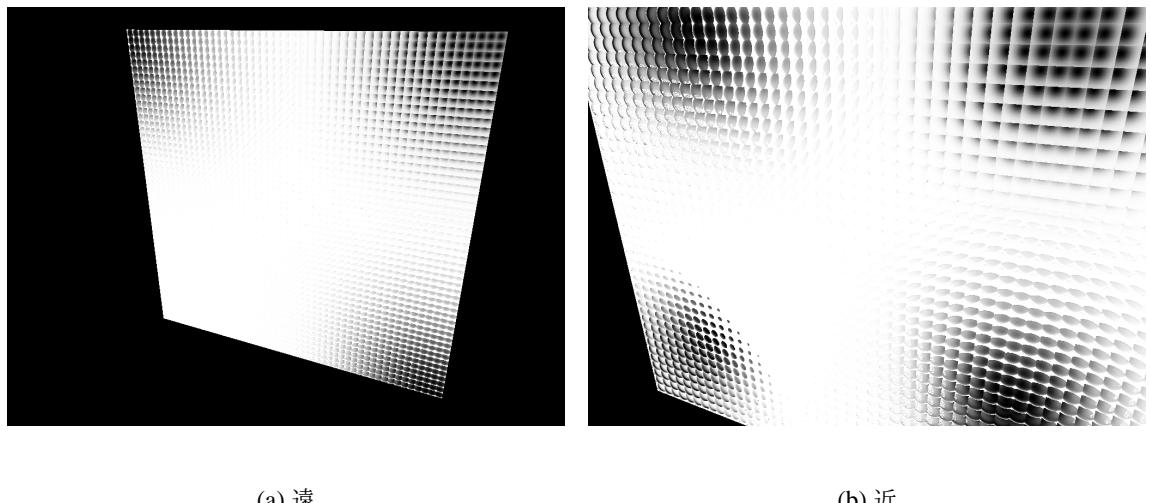


Fig. 6.13 視点距離と偽瞳孔の位置関係（平面オブジェクト）

(Fig.6.14(c)) は (Fig.6.14(a)) と (Fig.6.14(b)) の画像を個眼の大きさを一致させて重ねたものである。この画像や (Fig.6.13(a))(Fig.6.13(b)) の比較から、視点のとオブジェクトの距離が近づくほど偽瞳孔の相対的な距離が短くなっている、偽瞳孔の大きさが縮小していることが確認できる。ゆえに、本手法は第 4 章の結果と一致しており、特徴を再現できていると言える。

続いて、視点との距離と実行速度との関係について考察を行う。(Fig.6.15) に視点距離と実行速度の関係のグラフを示す。図番号は平面オブジェクトおよび球体オブジェクトが (Fig.6.16) および (Fig.6.17) に対応している。図番号が大きくなるにつれて、視点とオブジェクトとの距離は近づいている。グラフからは、視点距離が近づくとある程度までは実行速度が遅くなるが、さらに近づくと逆に実行速度がわずかに速くなっていることがわかる。これは、視点が近づくことによりオブジェクト表面に対して平行に入射する視線が増えるため隣接球処理 (6.4.1 節) を多く行うレンズが増え、結果的に処理速度が遅くなるが、一方で、視点がさらに近づくと画面内に表示されるオブジェクトが視点近傍の部分に限られるため隣接球処理を多く行わなくてはいけないレンズが画面内から減り、結果的にやや処理速度が速くなったものと考えられる。以上から、視点とオブジェクトとの距離は実行速度に少なからず影響を与える

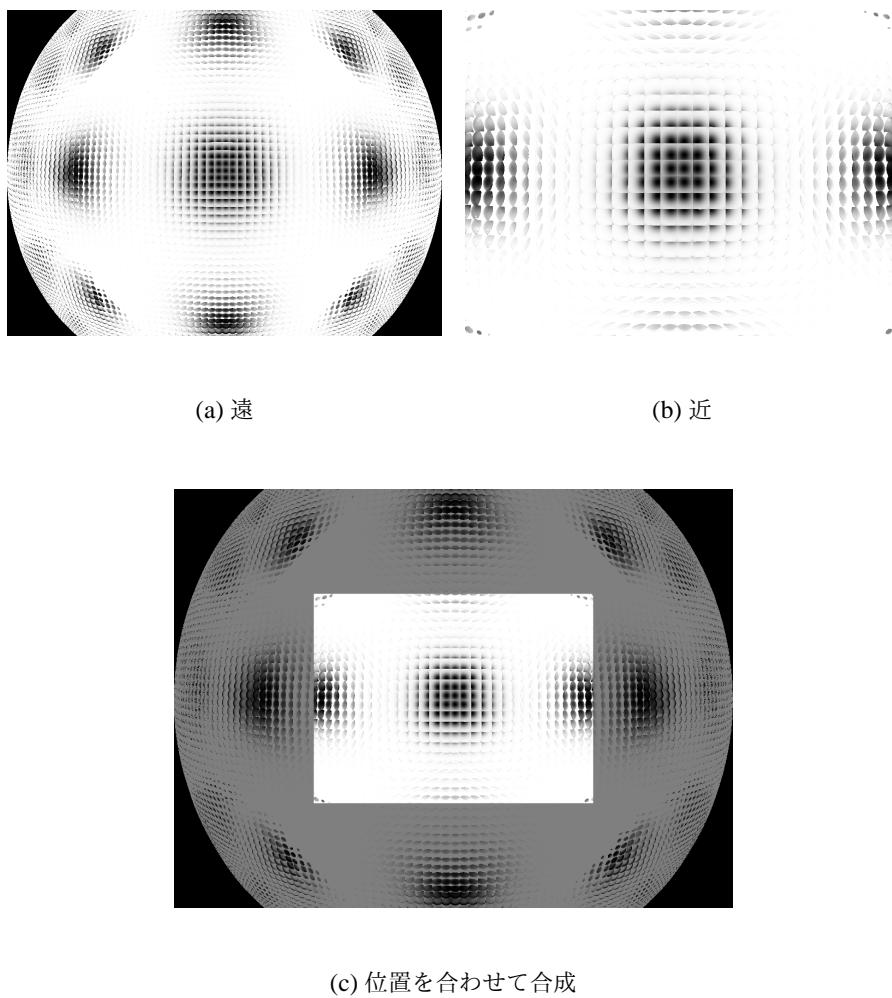
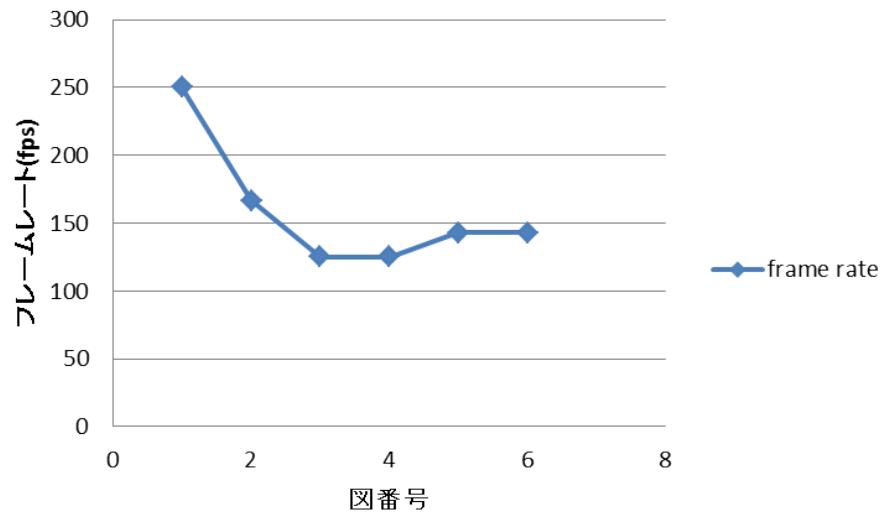
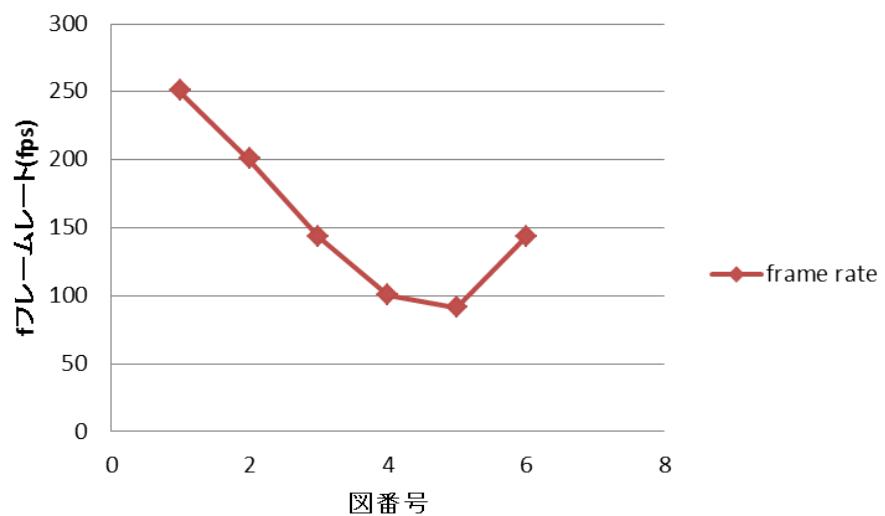


Fig. 6.14 視点距離と偽瞳孔の位置関係（球体オブジェクト）

が、隣接球処理の頻度による影響を間接的に受けているためであると言える。



(a) 平面オブジェクト



(b) 球体オブジェクト

Fig. 6.15 視点距離と実行速度の関係

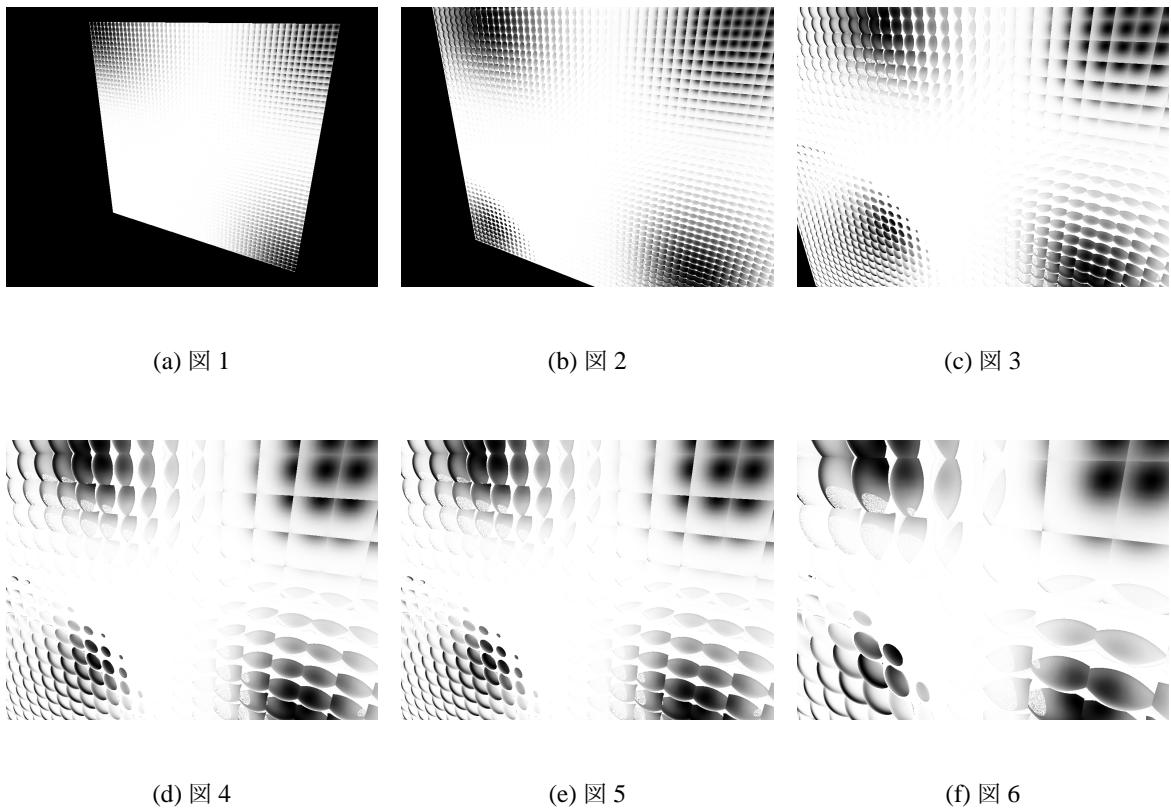


Fig. 6.16 視点距離と実行速度（平面）

6.4.3 個眼密度

5.3 節のテクスチャ解像度 R_t によって決定されるレンズの密度を個眼密度と呼称している。個眼密度は外観に大幅に影響を与えるため、各オブジェクトごとにユーザが適切な値を設定する必要がある。(Fig.3.2) に、 R_t の変化による外観の変化を示す。指定される球の半径 r は R_t に反比例する値をそれぞれ与えてある。また、 R_t のオーダーを変化させた場合、 R_t が大きく個眼密度が大きいほどわずかに実行速度が遅くなったものの、同一オーダー内であれば、実行速度に有意な差は見られなかった。

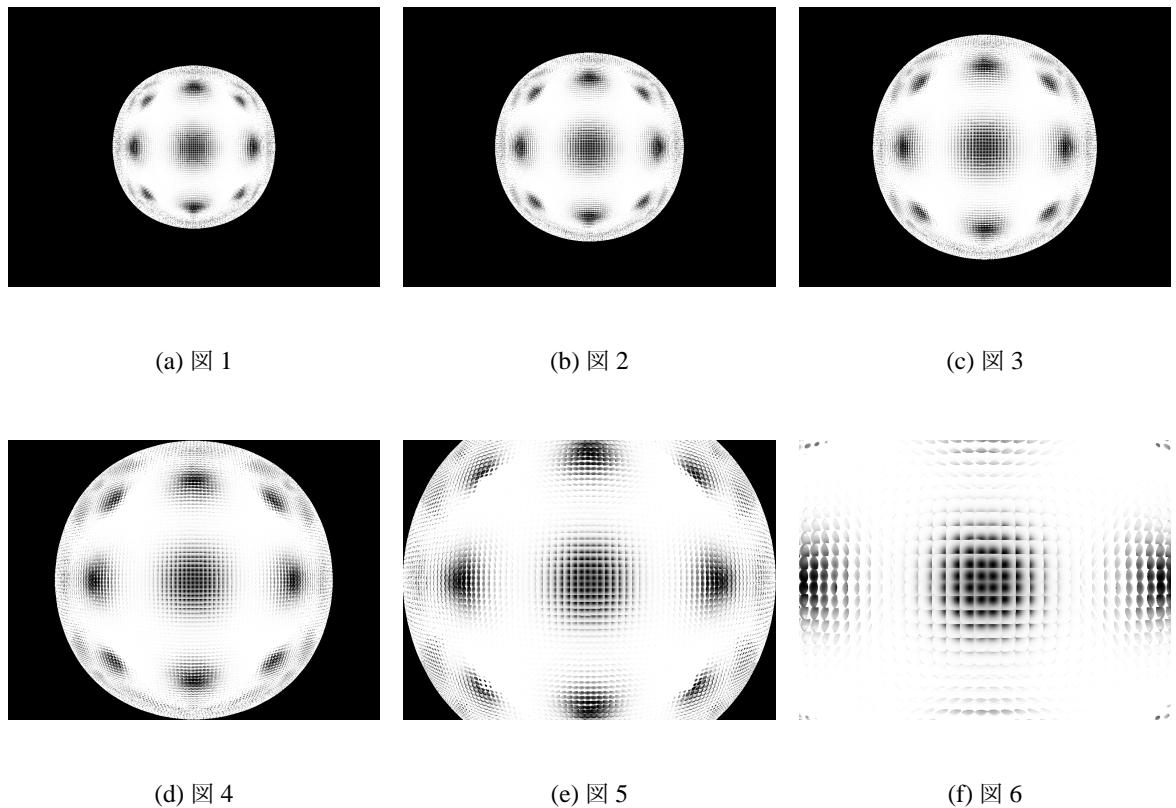


Fig. 6.17 視点距離と実行速度（球体）

6.5 実物との比較

昆虫の複眼は六角形の配置をしているため、3.3 節で述べたように side pupil は central pupil を中心として六方向に現れる。本研究では、簡単のために格子状の個眼配置をしているため、厳密には実物の配置とは異なる (Fig.5.4)。ゆえに、得られた結果では side pupil の数に違いが見られる。

また、球体オブジェクトに本手法を適用した例においては、視線方向と個眼配置を決定するテクスチャの軸方向との関係性により side pupil が 3 つ現れる場合と 4 つ現れる場合が確認できた。 (Fig.6.20(b))(Fig.6.20(d)) の赤い矢印がそれぞれの軸方向を表しており、結果画像 (Fig.6.20(a))(Fig.6.20(c)) では中心から軸方向に沿って side pupil が展開されていることがわかる。

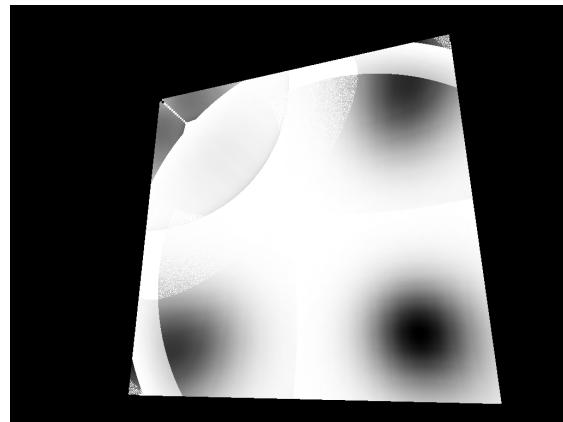
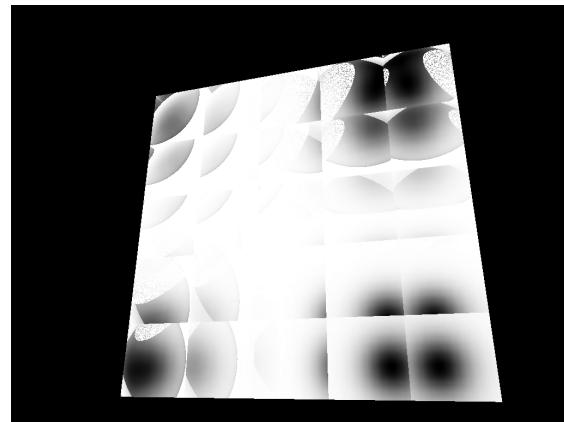
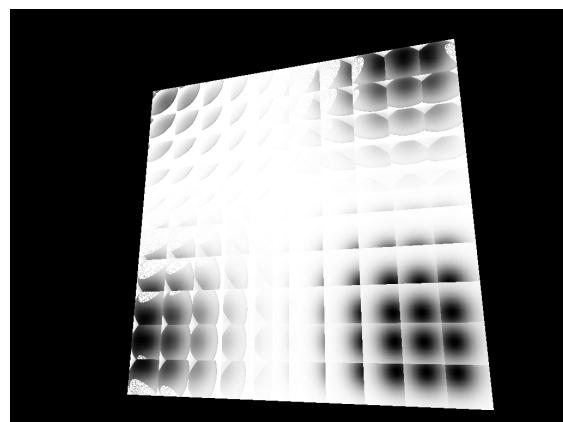
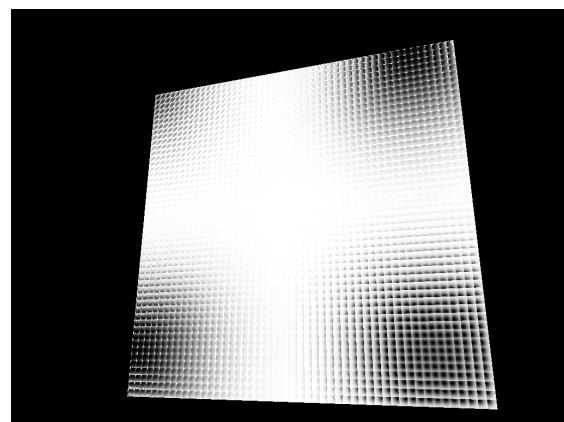
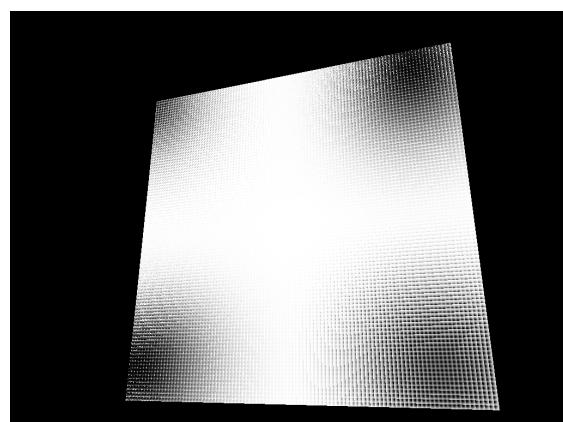
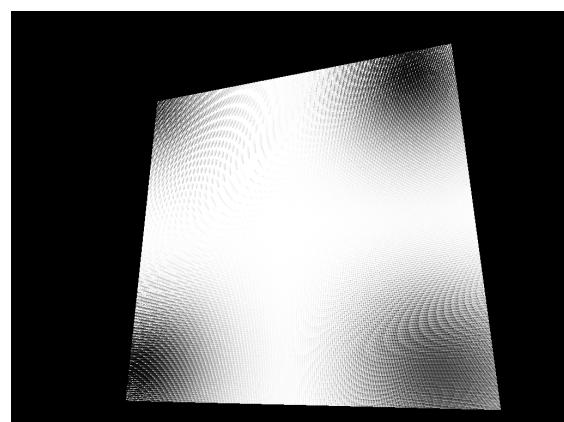
(a) $R_t = 1$ (b) $R_t = 5$ (c) $R_t = 10$ (d) $R_t = 50$ (e) $R_t = 100$ (f) $R_t = 500$

Fig. 6.18 個眼密度による実行結果の変化（平面）

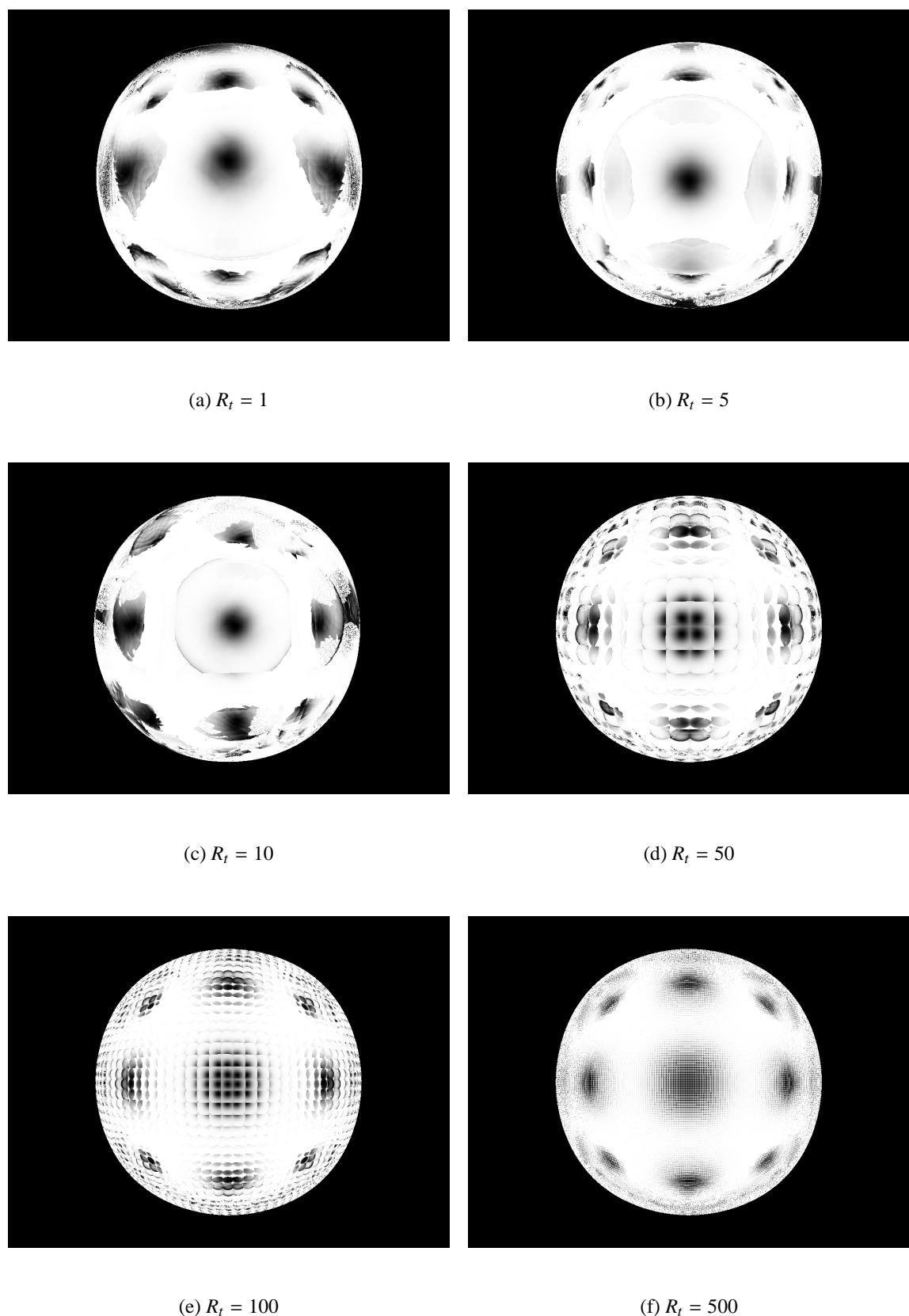


Fig. 6.19 個眼密度による実行結果の変化（球体）

る。偽瞳孔の位置関係は個眼の配置に依存しているので、実物の個眼と同様にレンズを配置すれば結果の再現性が高まると考えられる。

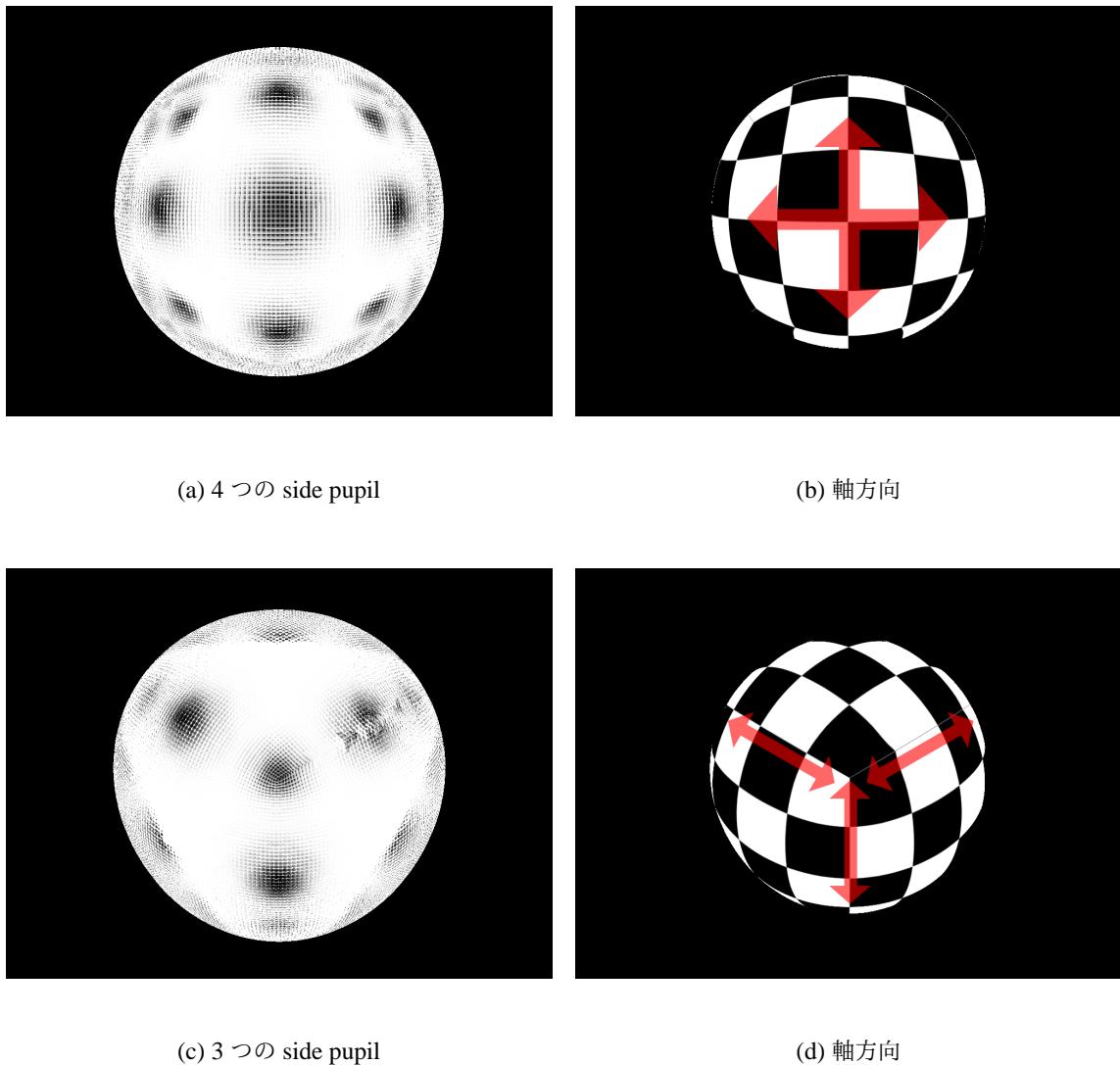
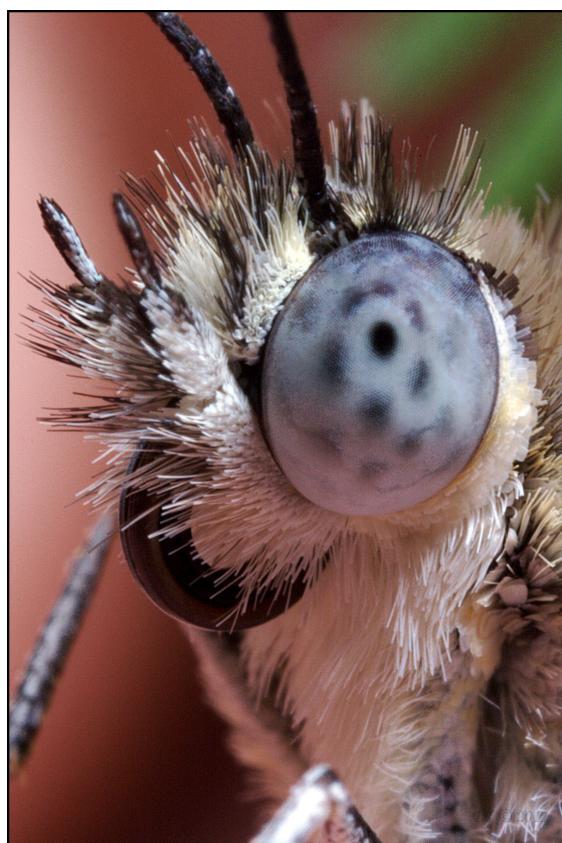
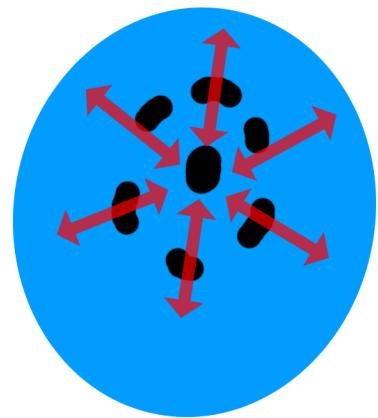


Fig. 6.20 個眼配置と偽瞳孔の関係

以上から、本手法によって得られた結果は偽瞳孔の現れ方や部分的に偽瞳孔の特徴をよく再現できているといえる。しかしながら、3.3.1節で述べたような色素細胞の位置によって偽瞳孔自体が六角形に近づく場合 (Fig.6.22) や、central pupil および side pupil の周囲にリング状にあらわれる偽瞳孔 ((Fig.6.23) 49) などの、本手法では表現できていない偽瞳孔現象についてはさらに試行の余地がある。



(a) 6つの side pupil([3] より転載)



(b) 軸方向

Fig. 6.21 実物の side pupil

6.6 さまざまなオブジェクト

本手法はさまざまなオブジェクトに対して適用可能である。以下にその例を示す。

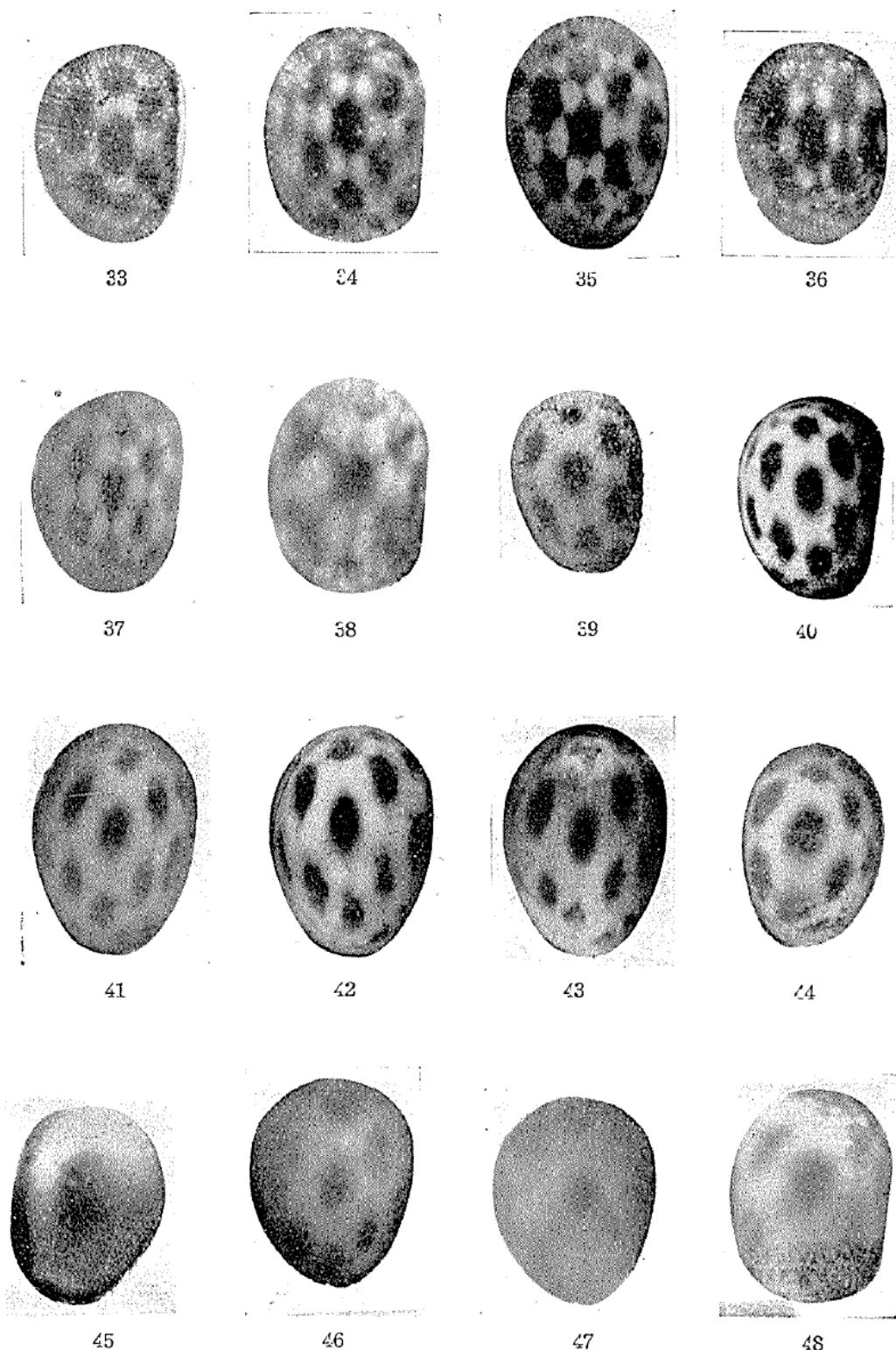


Fig. 6.22 六角形の偽瞳孔（写真:[4] より転載）

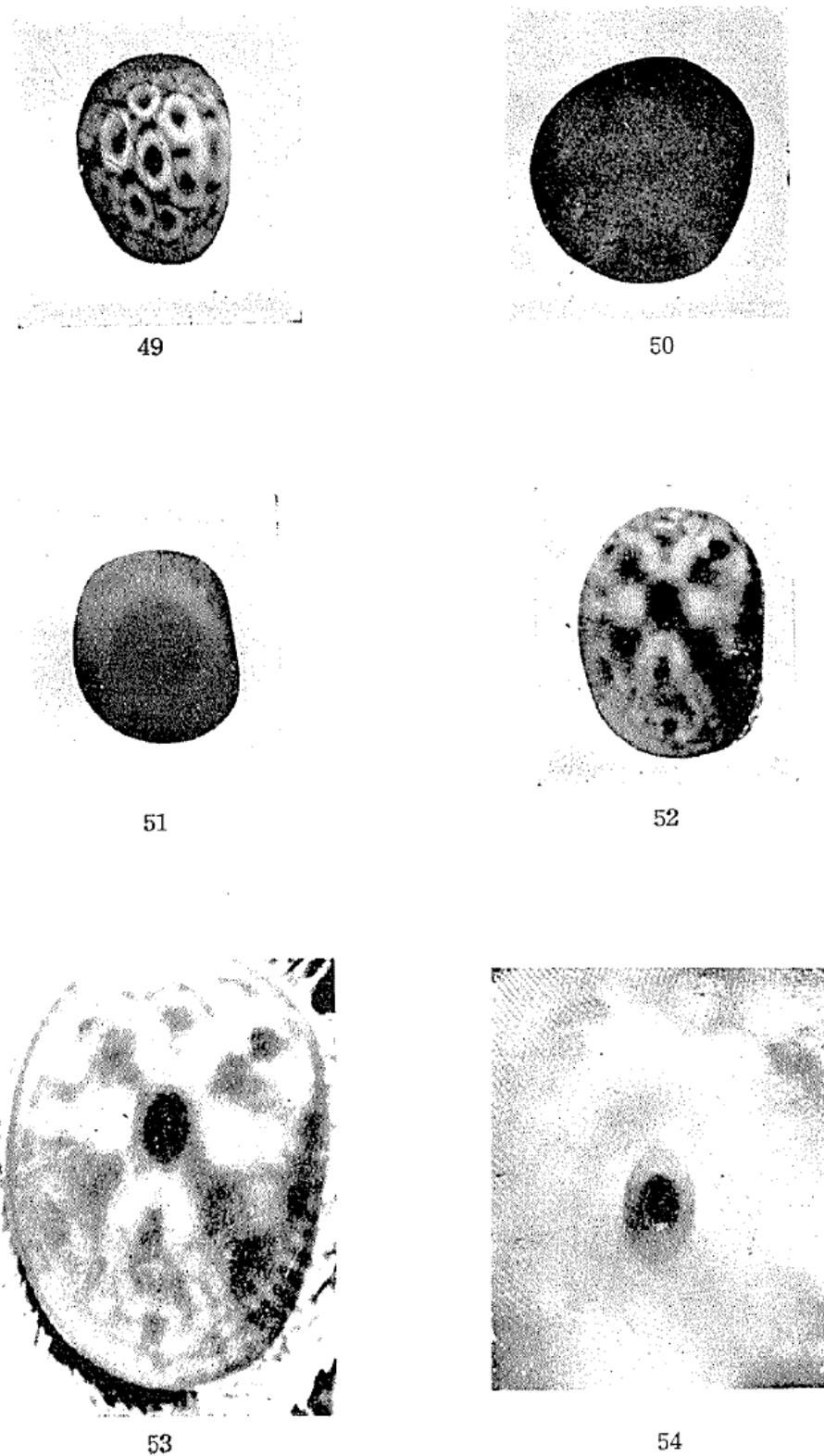
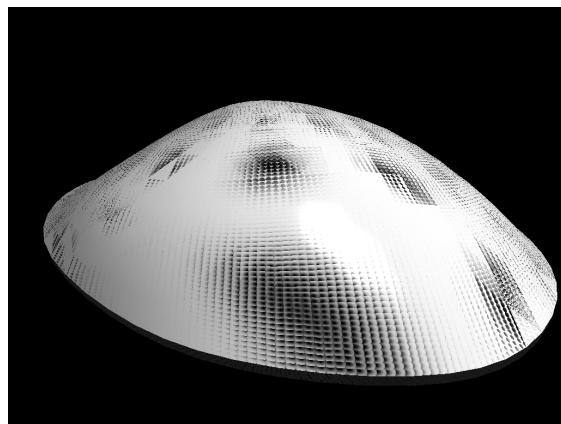
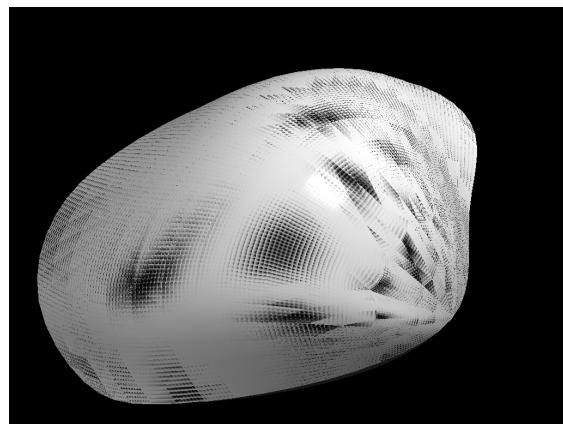


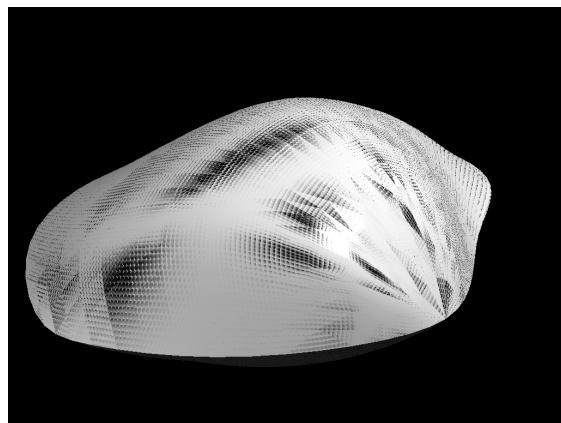
Fig. 6.23 リング状の偽瞳孔（写真:[4] より転載）



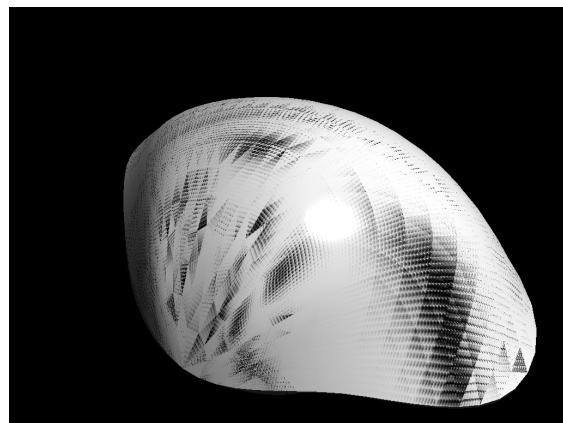
(a) 図 1



(b) 図 2



(c) 図 3



(d) 図 4

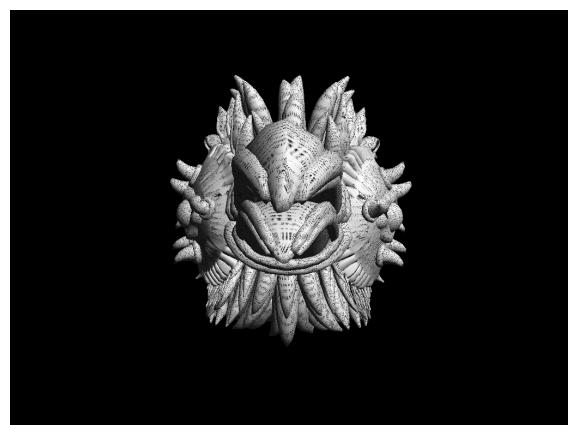
Fig. 6.24 貝殻オブジェクトへの適用

6.7 応用事例

本研究の応用事例として、2014年12月??日に行った「かごしまアートフェスタ2014」への展示の事例を紹介する。ここです！残ってるとこ



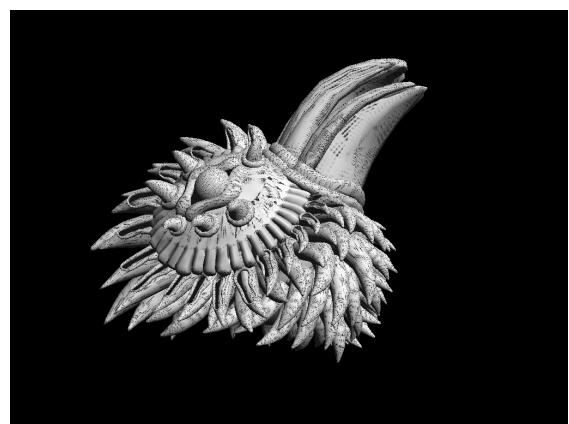
(a) 図 1



(b) 図 2



(c) 図 3



(d) 図 4

Fig. 6.25 鳥頭オブジェクトへの適用

第 7 章

おわりに

7.1 結論

本研究では、昆虫などの複眼に生じる偽瞳孔を CG によってリアルタイムに表現する手法を提案した。提案手法ではテクスチャと球体レンズによって複眼の光学的モデルを考案し、球体オブジェクトなどに対して手法を適用することで、複眼らしい偽瞳孔の模様を再現することができた。本研究の提案手法について以下の成果を確認できた。

- 光の屈折を含む内部構造を考慮したモデルによって central pupil, side pupil, second side pupil といった偽瞳孔を再現できる。
- 個眼レンズを複数通過する光の屈折をリアルタイム処理することができる。
- 任意のオブジェクト形状に対して複眼シェーダを適用することができる。
- さまざまなテクスチャを用いることで外観を変えることができ、アート作品への応用として展示を行った。

7.2 今後の展望

本手法では一般的な六角形の個眼の配置ではなく、格子状の配置を用いている。そのため、3つもしくは4つの side pupil を表現することはできたが、八木 [4] の記述にもあった6つの side pupil を表現することはできていない。本手法ではテクスチャ利用した球レンズの配置アルゴリズムを用いていたが、三角形ポリゴンのジオメトリ情報を利用した配置アルゴリズムを用いることで改良が可能であると考えられる。正三角形に近い三角形ポリゴンによって複眼の曲面を構成した場合、各頂点を球レンズの中心座標として用いることができる。また、この方法を用いれば個眼の配置だけならば実物を比較的正確に再現することがすでに可能となっており [1]、頂点位置に着目すれば良いため色素細胞のテクスチャとの同期も容易である。しかしながら、ポリゴン毎ではなく、ひとつの頂点から適用オブジェクト全体を構成するの頂点、面、法線方向などの情報へのアクセスが必要となるため、リアルタイム処理を達成するために全情報をバッファとしてあらかじめ GPU へと送信し保存するなどの工夫が必要だと考えられる。さらに、6.5節で述べたリング状の偽瞳孔などの、個別の生物に特徴的な偽瞳孔についてはしきみの解明やモデル化のための幾何的解釈が求められる。

謝辞

本論文は多くの方々のご協力により書き上げることが出来ました。この場を借りて皆様に深く感謝の言葉を申し上げます。

指導教員である東京大学大学院情報学環の河口洋一郎先生には、学部4年生の卒業論文から研究室の学生として大変お世話になりました。思えば、本論文の研究対象として複眼を選んだのは、先生の昆虫や水生生物への熱い関心からヒントを頂いたからにほかなりません。私が途中で研究を諦めそうになった時も、先生は真剣に激励してくださいました。

米先生パート

岩澤先生パート

職員パート

先輩パート

同級生パート

後輩パート

その他

参考文献

- [1] Mathias Koehler. Making of 'upside down'. <http://www.optisch-edel.de/papers/upsidedown/>.
- [2] Martin Rump, Gero Müller, Ralf Sarlette, Dirk Koch, and Reinhard Klein. Photo-realistic rendering of metallic car paint from image-based measurements. *Computer Graphics Forum*, Vol. 27, No. 2, pp. 527–536, April 2008.
- [3] scribbly nut. Crazy bird. <http://www.flickr.com/photos/60007327@N00/6291023483>.
- [4] Nobumasa Yagi. Studies on the compound eyes of lepidoptera: 1. on the compound eyes of butterflies, especially on the pseudopupil and its meaning to the phylogeny of species. 1951.
- [5] 敏夫永田. ビー玉を利用したモアレ学習材の開発 (北海道支部特集). 物理教育, Vol. 56, No. 1, pp. 77–79, mar 2008.

1.5

以上

1 p ~ 92 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