

東京電機大学

博士論文

トーラス状不可能図形の描画のための
錯視立体表現モデルの研究

Modeling of Illusion Representation Models
for Depicting Toroidal Impossible Figures

2022年3月

東京電機大学大学院

先端科学技術研究科 情報通信メディア工学専攻

千葉 堯

研究指導教員 教授 高橋 時市郎

あらまし

不可能立体のように見える絵(不可能図形)を描く方法のひとつとして, 特定の視点から見たときに錯視によって不可能立体に見えるような立体, すなわち錯視立体を用いる方法がある. 3DCGで表現した錯視立体を視点に応じて再モデリングし, 不可能図形のアニメーションを実現する手法が既に提案されている. しかしながら, それらの手法は不可能図形の表示に特化していて, 一般的な3DCGモデルと共存することは十分に考慮されていなかった.

本論文では, レンダリング環境に依存しない3DCGモデルとして与えられる錯視立体と, その3DCGモデルを視点に応じて動的に再モデリングするアルゴリズムとから構成される, 錯視立体表現モデルを提案する. 本モデルは, 再モデリングアルゴリズムによって, 不可能立体を想起させるような奥行感を知覚させる形状, すなわち不可能図形として描画されるように常に変形できる. 本モデルの再モデリングアルゴリズムは, レンダリング・パイプラインに副作用を与えないため, 一般的な3DCGモデルと共存可能である.

さらに, 不可能図形の影の在り方について考察し, その考察に基づいた付影処理手法を提案する.

目次

あらし	i
第1章 序論	1
1.1 本研究の背景	1
1.2 本研究の目的	4
1.3 本研究の成果	5
1.4 本論文の構成	6
第2章 不可能図形と錯視立体表現モデル	7
2.1 対象とする不可能図形	7
2.2 一般的な3DCGモデルとの共存可能条件	11
第3章 従来手法における錯視立体表現モデル	13
3.1 部分立体の稜線を接続する手法	14
3.1.1 Khohらの手法	14
3.1.2 大和田らの手法	15
3.1.3 藤木らの手法	15
3.1.4 中津らの手法	15
3.2 曲面を変形する手法	17
3.3 奥行値を逆ソートする手法	18
3.4 従来手法の比較と一般的な3DCGモデルとの共存可能条件への適合性の考察	19
第4章 一般的な3DCGモデルと共存可能な錯視立体表現モデル	21
4.1 錯視立体表現モデルと座標系・投影法	22
4.2 錯視立体表現モデルの形状定義と再モデリング処理	24
4.2.1 錯視立体表現モデルの初期形状	25

4.2.2	再モデリング処理(1)：3D形状変形処理	26
4.3	錯視立体表現モデルの拡張	28
4.3.1	既存表現モデルで発生する問題	28
4.3.2	再モデリング処理(2)：奥行整合処理	30
4.4	拡張表現モデルの実装と描画結果	32
4.4.1	拡張表現モデルを用いた不可能図形の描画結果	32
4.4.2	汎用レンダリング環境下での実装	41
4.4.3	複雑な形状への応用・テクスチャマッピングの利用	45
第5章	不可能立体の付影処理	55
5.1	不可能立体の影の在り方と錯視立体の影の要件	56
5.2	Shadow Illusion Objects法の提案	58
5.3	Shadow Illusion Objectsを用いた付影処理の実装	60
5.4	付影処理の主観評価実験	62
5.4.1	主観評価実験環境	62
5.4.2	被験者と実験手順・条件	62
5.4.3	評価項目・評価方法	75
5.5	付影処理の考察	76
5.5.1	影が他の3DCGモデルとの共存感に及ぼす影響	76
5.5.2	錯視立体の影と不可能立体の関係の知覚	77
5.5.3	トリックアートの面白さへの影の寄与	77
5.5.4	考察のまとめ	78
第6章	結論	79
6.1	本論文のまとめ	79
6.2	今後の展開	81
6.2.1	視点の特異点の除去	81
6.2.2	別の種類の不可能図形での表現	82
6.2.3	付影処理の高度化	83
謝辞		85
参考文献		87

第1章

序論

1.1 本研究の背景

不可能図形というトリックアートがある。これは錯視図形の種類で、一見すると3次元立体として実現可能なようにイメージでき、存在感もあるが、実際には人間の視覚が解釈したとおりには実現不可能な錯視図形である。

杉原によれば、不可能立体とは「立体実現問題の答えがノーであるにもかかわらず、人の心に引き起こす立体構造に関する印象」^[1]であり、立体と名前が付いているがあくまで印象

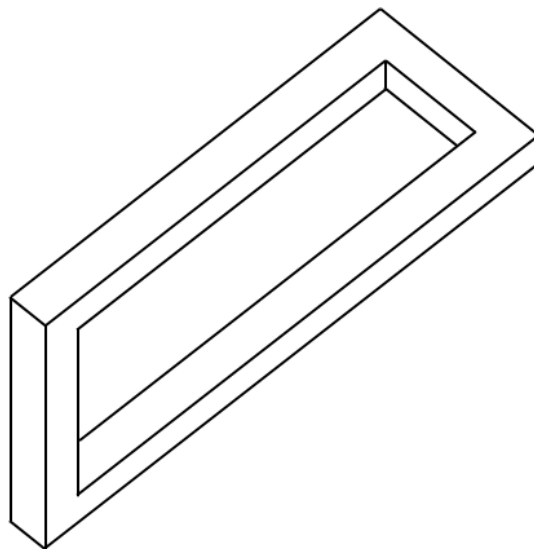


図1. ひねりを加えたトーラス状不可能図形^[1]

(または、印象上の立体)である。すなわち、ヒトが不可能図形を見ると、脳内では不可能立体をイメージするものの、その不可能立体を現実に作成することはできない。この矛盾が混乱や違和感を生み、不可能図形のトリックアートとしての面白さになっている。代表的な不可能図形のひとつが、ひねりを加えたトーラス状不可能図形であり(図1)、本論文では、単に**トーラス状不可能図形**と呼ぶ。トーラス状不可能図形をモチーフにした芸術作品は、図2に示すEscherの“Waterfall”^[2]をはじめ、多くの芸術家たちによってつくられ、人々を魅了している。

不可能図形の中には、ヒトの視覚が解釈した形状に限らなければ、3次元立体として実現可能な立体がある^[1]。そのような立体は、ヒトの解釈した形状とは大きく異なるが、ある特定の視点から見た場合、その立体は観察者の目には錯視によって不可能図形として映り、不可能立体としての印象を想起させる。この実現可能な立体を**錯視立体**と呼ぶ。錯視が成立し、錯視立体が不可能図形として知覚される特定の視点を、その錯視立体の**錯視成立点**と呼ぶ。錯視立体を錯視成立点以外を視点として見た場合、不可能図形と異なる形状として投影されるため、錯視効果は失われる。この視点を**錯視不成立点**と呼ぶ。

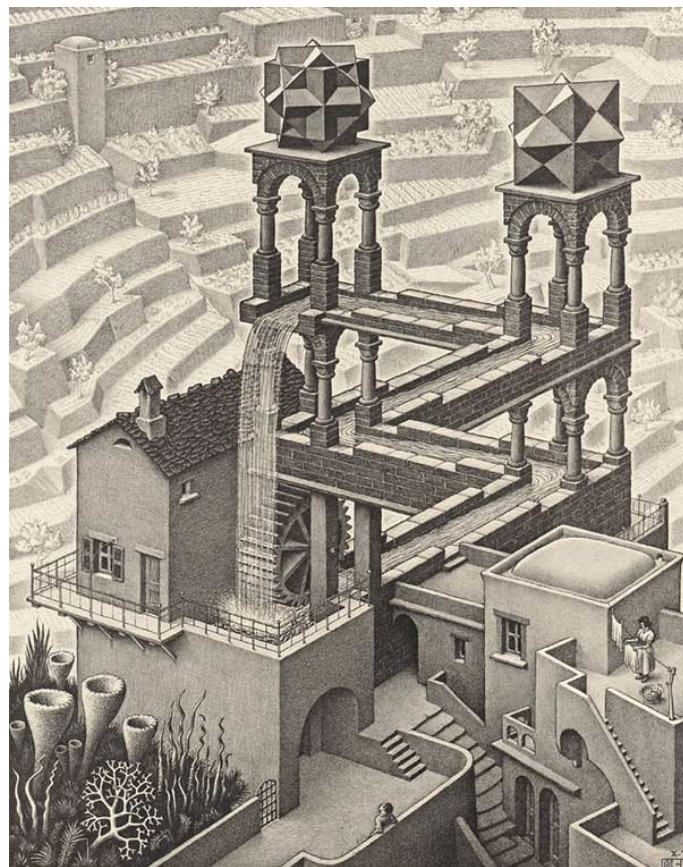


図2. 不可能図形をモチーフにした芸術作品^[2].

現実の錯視立体はそれぞれ特定の錯視成立点しか持つことができないが、錯視立体をコンピュータ上の3DCGモデルでモデリングしておき、観察者の視点が移動するたびに、錯視成立点が観察者の視点と同じ位置になるように、錯視立体の3D形状を動的に再モデリングすれば、不可能図形を3DCGでアニメーションさせることができる。本論文では、錯視立体の3DCGモデルと、視点の移動に合わせて錯視立体を動的に再モデリングするアルゴリズムとを合わせて、**錯視立体表現モデル**と呼ぶ。

近年、様々な錯視立体表現モデルが提案され、不可能図形のアニメーション手法が報告されている^{[3][4][5][6][7][8][9][10]}。しかし、従来手法では、最も簡単なトラス状不可能図形であっても、特殊なモデリングツールやレンダリング環境、あるいは研究者が独自に考えたアルゴリズムを直接実装した専用レンダラでしか、不可能図形を描画する術が用意されていないことが多い。錯視立体表現モデルが特殊なレンダリング環境に依存していると、研究者ではない3DCGアーティストなどが実装することは難しい。3DCGを用いた作品中で不可能図形アニメーションを表示できるようにするためには、汎用レンダリング環境で実装可能とすることが求められる。

本論文では、汎用レンダリング環境の例として、Unity^[11]を使用する。Unityは代表的なゲームエンジンのひとつで、3DCGのレンダリング環境と、3DCGを用いた作品を製作するための統合開発環境からなるソフトウェアである。昨今進歩著しいUnityは、ゲームエンジンとして知られてはいるものの、単にゲームを作成するのみならず、映像作品の製作や、AR/VR等のインタラクティブコンテンツの製作に広く用いられている。そのため、錯視立体表現モデルがUnity上で実装可能となれば、急速に広がりを見せる3DCGを用いたコンテンツの多くで不可能図形アニメーションが表示可能であると言える。

1.2 本研究の目的

本研究では、錯視立体の再モデリング処理を、レンダリング・パイプラインに副作用を与えないように実現することで、一般的な3DCGモデルと共存可能な錯視立体表現モデルを提案する。提案モデルは、独自実装のレンダラではなく、汎用レンダリング環境で不可能図形を描画可能であり、なおかつ、他の3DCGモデルと共存可能であるため、3DCG作品中に不可能図形のアニメーションを容易に組み込むことができるようになる。

さらに、錯視立体と他の3DCGモデルが共存するシーンで付影処理ができれば、両者が自然に共存しているような映像表現が可能となる。Unityをはじめとする汎用レンダリング環境では、付影処理が標準実装されているが、これは、一般的な3DCGモデルを対象としたものである。不可能立体は印象上にしか存在しない立体であり、不可能立体の影はどうあるべきか、今まで検討されていない。そこで、不可能立体の影の在り方についての考察も踏まえて、新たな付影処理手法を提案する。主観評価実験を行い、提案した付影処理手法の妥当性を検証する。

1.3 本研究の成果

本研究により得られた成果は以下の通りである。

(1) 一般的な3DCGモデルと共存可能な錯視立体表現モデル

錯視立体の3DCGモデルと、視点の移動に合わせて錯視立体を動的に再モデリングするアルゴリズムからなる、錯視立体表現モデルを提案した。本モデルをレンダリングすることで、不可能立体に見える図形、すなわち不可能図形を描くことができる。本モデルは、一般的な3DCGモデルと共存可能である。本モデルを用いることで、汎用レンダリング環境で他の一般的な3DCGモデルとともにレンダリングすることが可能となった。

(2) 不可能立体アニメーションにおける付影処理

不可能立体の影の在り方について考察し、錯視立体表現モデルの付影処理手法を提案した。主観評価実験により、考案した付影処理手法の妥当性・有用性を実証した。

1.4 本論文の構成

本論文の構成を以下に記す。

第1章 序論

本研究の背景，目的，成果について概説する。

第2章 不可能図形と錯視立体表現モデル

本研究で対象とする図形について述べる。

第3章 従来手法における錯視立体表現モデル

不可能図形を CG でレンダリングする関連手法を調査・分析した結果についてまとめる。

第4章 一般的な3DCGモデルと共存可能な錯視立体表現モデル

本論文で提案する錯視立体表現モデルについて述べる。

提案した錯視立体表現モデルを汎用レンダリング環境へ実装した結果についても述べる。

第5章 不可能立体の付影処理

不可能立体の影の在り方を考察し，錯視立体表現モデルの付影処理手法を提案する。

その妥当性・有用性についても述べる。

第6章 結論

本研究の内容，成果，今後に残された課題についてまとめ，本論文の結言とする。

第2章

不可能図形と錯視立体表現モデル

2.1 対象とする不可能図形

杉原^[1]によれば、長方形の断面をもった四角柱を環状に組み合わせてつくるトーラスに、ひねりを加えることでつくられる不可能図形のグループがある。本論文では、単にトーラス状不可能図形と呼ぶ。図3に3種類のトーラス状不可能図形の例を示す。図3に挙げたトーラス状不可能図形は、シンプルな形状をしているが、立体的に捉えようとするとき奥行きを誤認するという錯視によって成り立っている。また、縦・横・奥行の3次元方向に柱が伸びている形状は応用の幅が広く、Escherの作品Waterfall^[2]も基本的な形状は同じであるなど、芸術作品

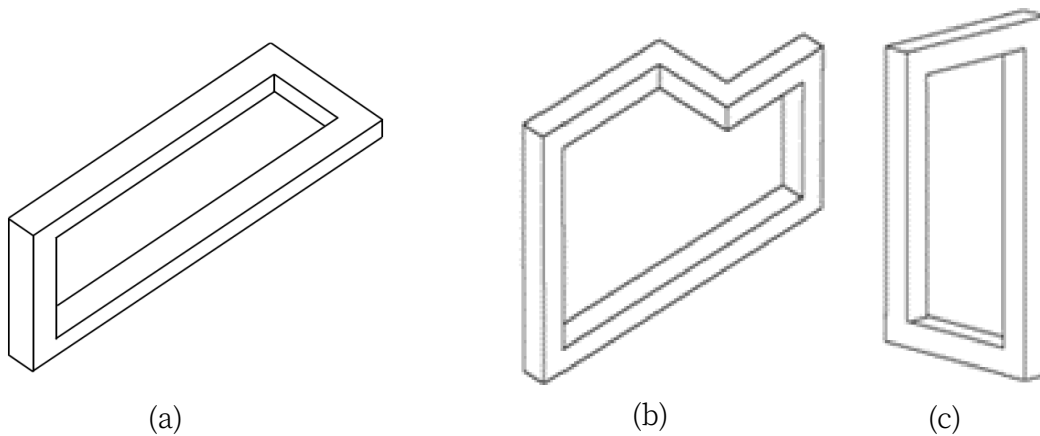


図3. ひねりを加えたトーラス状不可能図形

としての利用例も多い。

トーラス状不可能図形と同じ仕組みの錯視の仕組みには、無限にループする水路やペンローズの階段のようなモデルが知られている。先行研究結果^{[3][4]}によれば、本論文で提案する手法は、トーラス状不可能図形のみならず、それらのモデルに対しても、適用可能である。一方、図4に示すImpossible trident^[12](図4の引用元^[13]ではBlivetと呼ばれているが、同一の不可能図形である)のような、全く異なる錯視の仕組みを用いる不可能図形は表現することができない。

本論文では、このようにシンプルかつ応用例の多いトーラス状不可能図形(図1)を描画対象とする。トーラス状不可能図形を描画する錯視立体はいくつかの種類がある。その一例を図5に示す。この立体は、図6に示すように、錯視成立点Aから見たときにのみ、図6(a)のようにトーラス状不可能図形として見える。視点を点Bに移動すると、図6(b)のように見えて、不可能図形として知覚されなくなる。

物理的に実在する現実の錯視立体は錯視成立点が固定されている。しかしながら、本論文で提案する錯視立体表現モデルは、3DCGで描画される映像であるため、視点を変更されて

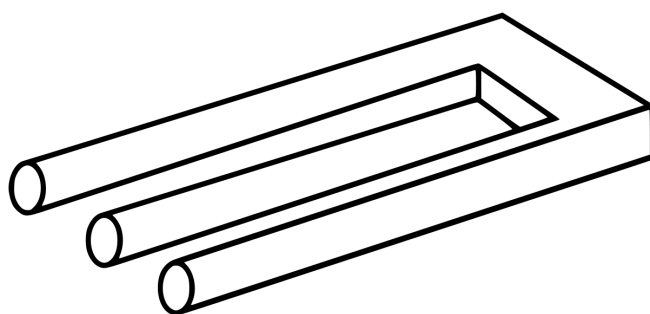


図4. Impossible trident^[13]

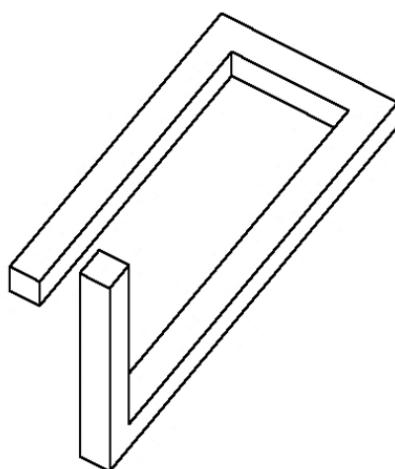
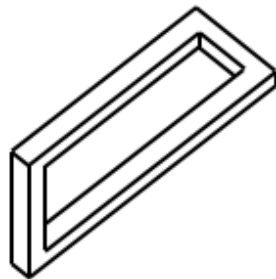
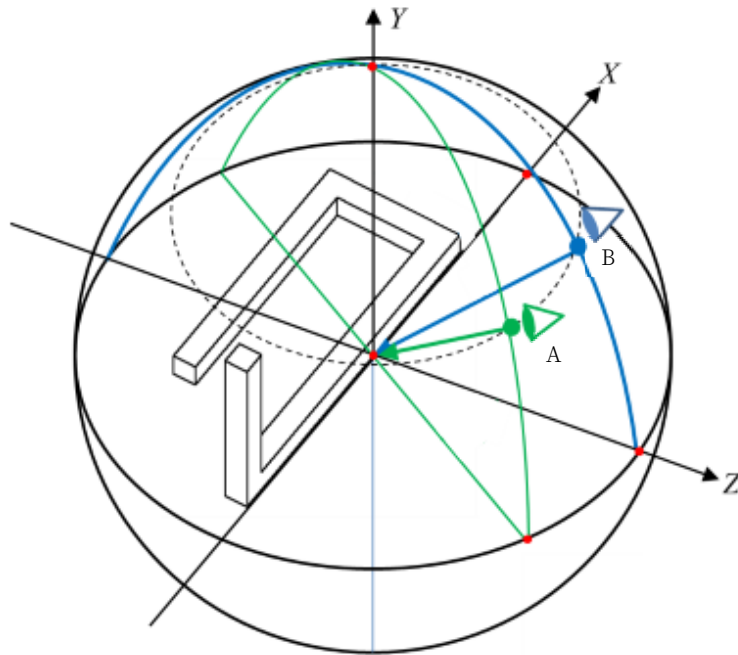
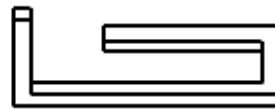


図5. 錯視立体の例

も、その視点が錯視成立点となるよう常にモデルを変形できる。この処理を、**錯視成立点の変更**あるいは**錯視成立点変更処理／アルゴリズム**と呼ぶ。通常、錯視成立点を変更すると、もとの錯視成立点は錯視成立点ではなくなる。図6 (b)は、錯視立体表現モデルの錯視立体の3D形状を、錯視不成立点から、再モデリングするアルゴリズムを適用せずに描画した図である。



(a) 錯視成立点Aから
みた錯視立体



(b) 錯視不成立点Bから
みた錯視立体

図6. 視点による錯視立体モデルの見え方の違い

2.2 一般的な3DCGモデルとの共存可能条件

これまでに提案された不可能図形描画手法は、その錯視立体表現モデルの3D形状、3DCGモデルの種類、錯視成立点変更アルゴリズムなどがそれぞれ異なるだけでなく、描画に用いるレンダラの実装まで扱ったものなど、種々様々である^{[3][4][5][6][7][8][9][10]}。しかし、他の3DCGモデルと共存可能とするためには、単に不可能図形を描画できるだけでは不十分である。錯視立体表現モデルが、他の3DCGモデルと共存可能であるための条件を①～⑤に整理する。

① サーフェスモデルであること

ゲームエンジンに代表される汎用レンダリング環境では、一般にサーフェスモデルが用いられる。錯視立体表現モデルが3DCGモデルと共存するためには、錯視立体表現モデルの3D形状もまた、サーフェスモデルでモデリングされるべきである。

② 写実的なレンダリング機能を有すること

錯視立体表現モデルが作品中の他の3DCGモデルと共存するためには、他の物体と同等にレンダリングされる必要がある。したがって、線画表現だけでなく、シェーディングやマッピングを含む写実的なレンダリングが可能であることが必要不可欠である。

③ 汎用レンダリング環境で実装可能であること

錯視立体表現モデルと他の3DCGモデルを共存させて描画するには、条件①②だけでは必要十分とは言えない。錯視立体に特化して研究者がアルゴリズムを直接実装したレンダラに、他の3DCGモデルのレンダリング機能を持たせるのは非常に困難である。錯視立体表現モデルは、ゲームエンジンなどの汎用レンダリング環境下で実装される必要がある。

逆に、汎用レンダリング環境では、一般的なサーフェスモデルを写実的にレンダリング可能なので、条件③を満たせば、条件①②は自動的に満たされる。

④ シンプルかつ自動化可能な錯視の仕組みであること

錯視成立点変更アルゴリズムがシンプルであり、レンダリング・パイプラインに副作用を及ぼさないことが重要である。副作用のあるアルゴリズムを汎用レンダリング環境に実装することは、見通しを悪くする。また、錯視を表現するための微妙なパラメータ調整や複雑な設定など、多くの手作業を伴う仕組みは極力なくし、自動化されているべきである。

⑤ 不可能立体の印象がロバスト性を有すること

錯視成立点変更アルゴリズムによって、錯視立体表現モデルの3D形状を变形するが、錯視立体としての錯視が成立する状態を継続したまま錯視成立点の変更が可能な範囲が広いアル

ゴリズムを採用すべきである。かつ、その変形はなるべく自然な印象を与え、変形の前後で見ているものが同一の不可能立体で、スムーズに変形しているという印象を与えるべきである。これを本論文では不可能立体の**印象のロバスト性**と呼ぶ。

第3章

従来手法における錯視立体表現モデル

本章では、3.1節で、トーラス状不可能図形を描画するためにこれまでに提案された代表的な錯視立体表現モデルを概説する。次に、それらが前章の条件①～⑤にどの程度合致しているかを、3.2節で検討する。

なお、文献によっては不可能図形(impossible figures)や錯視立体(optical illusion objects)の定義が本論文と異なる場合や、どちらかを不可能立体(impossible objects)と表記している場合がある。また、不可能立体、錯視立体、不可能図形の区別をせずに、すべてをまとめて不可能立体としている場合もある。本論文では杉原の定義^[1]に従い、実現不可能な立体の印象のみを**不可能立体**とよび、画面に投影された2次元図形を**不可能図形**、特定の視点からの投影図が不可能図形になる3D立体を**錯視立体**と記す。

3.1 部分立体の稜線を接続する手法

Khohら^[5] および大和田ら^[6] は、矛盾のない部分立体の稜線を接続することによって不可能図形を表現する手法を考案した。これらの手法により、不可能図形が回転するアニメーションが作成された。

藤木ら^{[7][8]} および中津ら^[3] は、錯視立体を角柱の集合として与える。トーラス状の不可能図形を、前章で述べた錯視立体を用いて表現する。錯視立体には、角柱同士の間が接続されていない部分が存在する。そこで、離れている角柱同士を投影図上で接続しているように知覚させるために、視点に応じて各角柱を伸縮および移動する手法を提案している。

3.1.1 Khohらの手法

Khohらの手法^[5] では、点対称な形状の途中にひねりが加わった形状の不可能図形をレンダリングしている(図7)。しかし、彼らの手法では、点対称に反転させた2つのオブジェクトを接続しているため、接続する面同士の幅が同じ幅になるように変形させないといけないという制約がある。そのため、ある視点では正方形の断面をもつ角柱でできているように見えた不可能図形が、別の視点では面に見えるほど薄くなってしまう。視点位置によって想起される不可能立体の形状が大きく異なるため、同じ錯視立体が変形していると知覚されにくいという問題点がある。

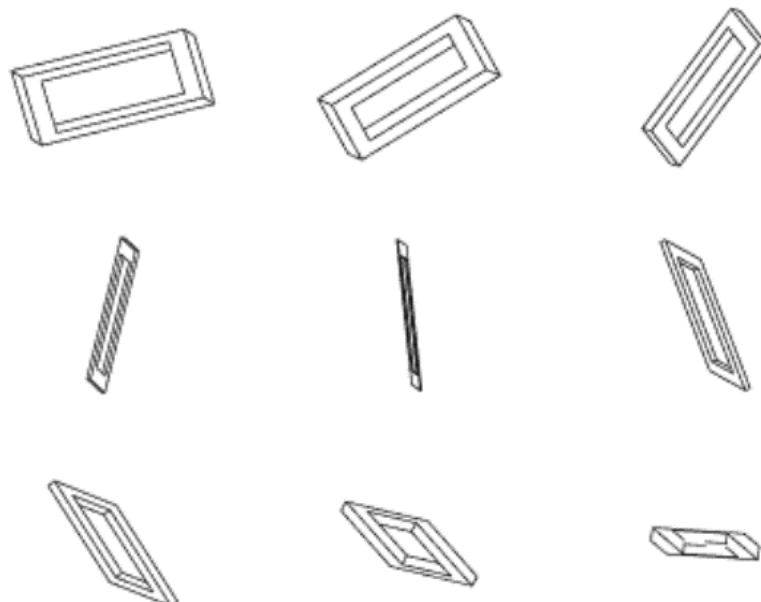


図7. Khohら^[5]の手法による不可能図形の出力結果の例

3.1.2 大和田らの手法

大和田らの手法^[6]では、任意の数・形状の部分立体を用いて不可能図形のための接続を作成することができる。しかし、彼らの手法は、隠面消去後に見えている稜線の接続を、同じ組の面が描画されている間に限り維持する手法であるため、例えば錯視立体の周りを1周するような動作は行うことができない。

3.1.3 藤木らの手法

藤木らの手法^{[7][8]}では、錯視立体を角柱の集合として与える。彼らは、離れている角柱同士を投影図上で接続しているように知覚させるために、視点に応じて各角柱を伸縮および移動する手法を提案した。これにより、角柱同士の間で接続していない部分が接続しているように知覚され、錯視立体となることで、トーラス状不可能図形を描画可能とした。しかしながら、彼らの手法には、水平方向に視点移動すると、垂直方向にも面が移動するため、元の不可能図形が与える印象とは異なる形状になってしまうという問題点があった。

3.1.4 中津らの手法

中津ら^[3]は、錯視成立点を非常に大きく離れた箇所に変更しても、より自然な印象を与える、トーラス状不可能図形の錯視立体表現モデルを提案した。図8は、視点の緯度 $\phi = 5\pi/18$ を固定し、経度 $\lambda \in [-\pi, \pi)$ を $\pi/6$ ずつ移動させたときの不可能図形の描画結果である。彼らが扱う錯視立体は、図6のように、幾何学的には離れている角柱同士の稜線が、投影図上で接続しているように見せる立体である。そのために、視点に応じて角柱を伸縮させる再モデリングアルゴリズムを提案している。

中津らのモデルは、角柱の伸縮量を微調整する仕組みを取り入れている。そのため、図8に示すように、錯視成立点の変更によって錯視立体を再モデリングしても、不可能立体の印象を大きく損なわずに、不可能図形を表示できるという長所がある。また、錯視成立点の変更可能範囲も、極付近を除いた天球全体と、非常に広い(条件④⑤適合)。

この手法では、錯視立体をサーフェスモデルでモデリングしているが、その不可能図形は線画としてのみレンダリングされる。また、錯視成立点の変更に対して錯視立体の3D形状の再モデリング処理とレンダリング処理とが分離していない問題がある(条件①②③不適合)。

なお、本節で紹介した手法^{[3][5][6][7][8]}は、すべて不可能図形を線画でレンダリングすることを前提としており、前章の共存可能条件①を満たしていない。

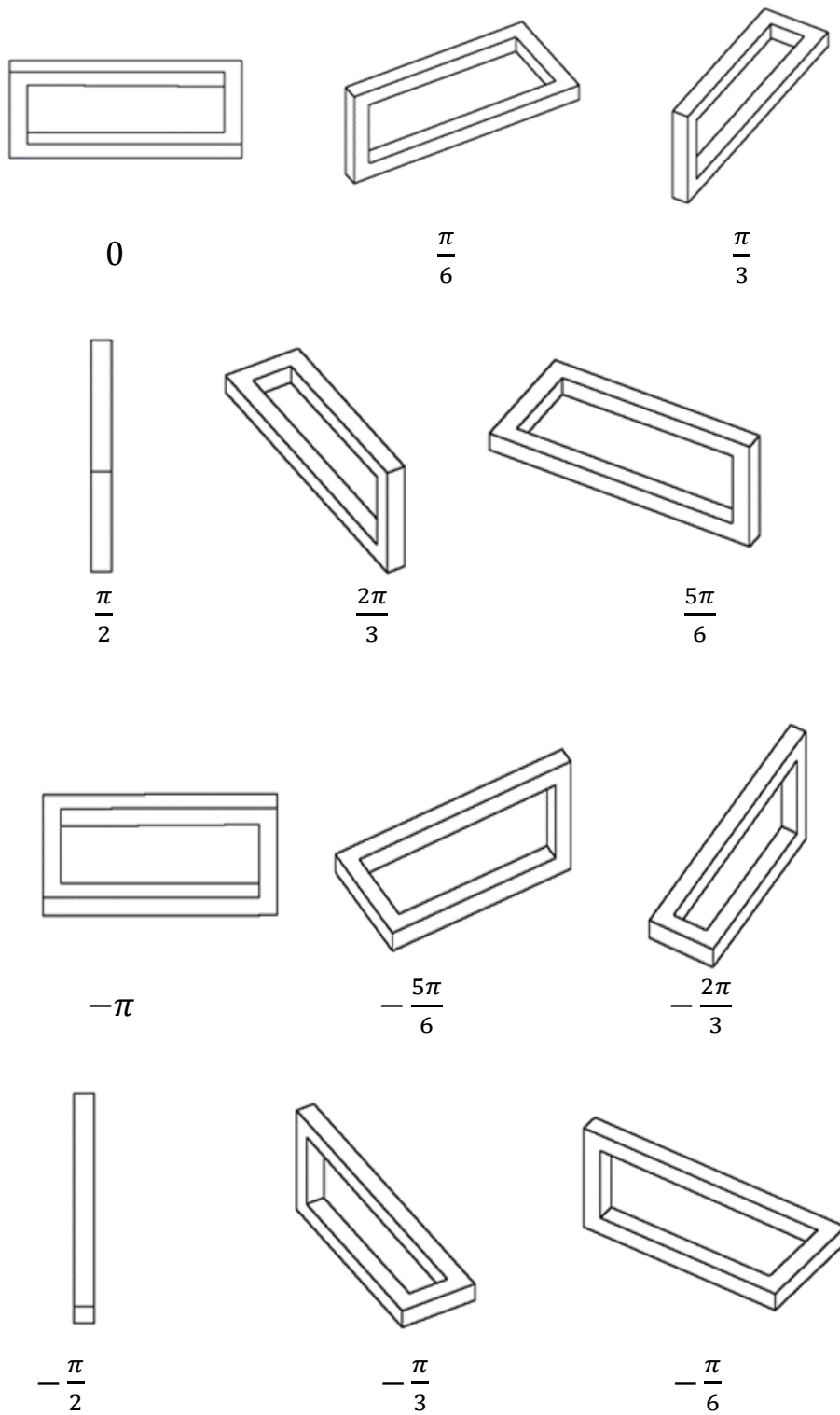


図8. 伸縮する角柱による不可能図形の描画結果^[3]

3.2 曲面を变形する手法

Wuら^[4]は、Thin-Plate Spline曲面で構成された錯視立体を用いた。錯視立体の再モデリングに際して、構成する曲面の変形を抑えて平面に見せるように、かつ描画結果の不可能図形の稜線の平行関係を保持するような、錯視立体表現モデルを提案した。この手法によって、図9に示すように、ひねりを加えたトーラス状不可能図形を含む複数の不可能図形を3DCGで表現した。また、視点をある限定された範囲で動かすことも実現した。

錯視立体はThin-Plate Spline曲面で構成されているので、シェーディング処理なども可能である(条件①②適合)。しかし、この手法では、中核となっているThin-Plate Spline曲面処理のために、特殊な専用レンダラを開発しており、汎用性・拡張性に乏しい(条件③不適合)。

また、見えている面同士の遮蔽関係が変化しない間しか、錯視効果が発生させる稜線の平行関係を保持できないため、錯視成立点の変更可能範囲が狭いという問題がある。面同士の遮蔽関係が変化した場合、新たに錯視立体の稜線の平行関係を手動で設定する必要がある。この作業は極めて煩雑である上、不可能立体の印象のロバスト性を損なう原因ともなっている(条件④⑤不適合)。

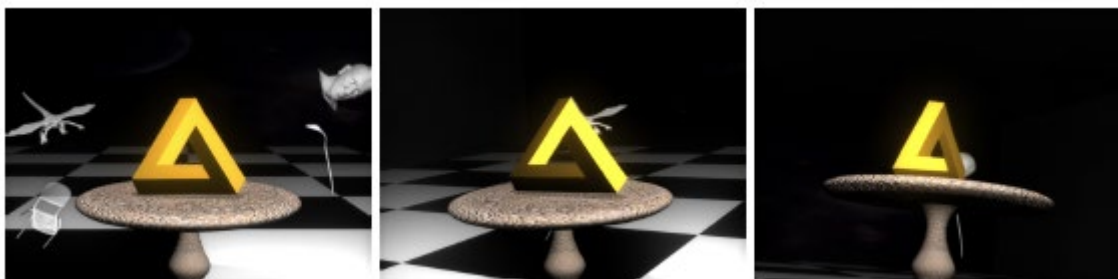


図9. Wuら^[4]の出力結果の例

3.3 奥行き値を逆ソートする手法

篠原ら^[9]はRay Tracing法，吉川ら^[10]はZ-buffer法をそれぞれ用いて，レンダリング時の奥行き値を逆ソートすることによって，不可能図形を表示する手法を提案した．この2つの手法に共通の利点は，一般的なRay TracingやZ-buffer法を用いたレンダラの奥行き判定処理のみを逆転させればよい点である．シェーディングやマッピングの他，反射・屈折も表現可能である(条件①②適合)．

しかし，奥行き値を逆ソートする範囲を，視点を移動する都度，錯視立体ごとにそれぞれ手作業で指定する必要がある，極めて煩雑である．特に，アニメーション生成の際には毎フレーム手作業となり，さらに煩雑である (条件④⑤不適合)．

また，奥行き判定を逆転させる処理を組み込んだレンダラを自前で実装する必要があり，汎用レンダリング環境で動作するとは言えない (条件③不適合)．

3.4 従来手法の比較と一般的な3DCGモデルとの共存可能条件への適合性の考察

本章で紹介した従来手法のうち、代表的な手法として、中津ら^[3]、Wuら^[4]、篠原ら^[9]、吉川ら^[10]の手法を選び、それらの手法が、前章2.2節で述べた通常の3DCGモデルとの共存可能条件①～⑤にどれくらい合致しているかを検討した。同時に、それらの手法を汎用レンダリング環境に移植する難易性も検討した。検討結果を要約して表1に示す。

(1) 中津らの手法について

中津らの手法^[3]は、条件④⑤を満たしている。錯視を実現する仕組みがシンプルで、手作業が入ることなく、不可能図形を自動的に表示可能である。錯視成立点を変更する際の変更可能範囲が広く、様々な視点から同じ不可能立体として知覚されるという利点がある。錯視立体の再モデリング処理とレンダリング処理が互いに独立に動作することから、汎用レンダリング環境への移植は比較的容易である。

中津らの手法の弱点は、線画での不可能図形表現に止まっている点である。それは線画での錯視表現を優先するために、サーフェスモデルの機能を使わなかったからである。

(2) 他の手法について

一方、他の手法(Wuら^[4]、篠原ら^[9]、吉川ら^[10]の手法)は、非常に巧妙精緻、高機能かつ複雑なアルゴリズムにより錯視が実現されている。それゆえに、いずれの手法も煩雑な手作業が必要である。また、錯視成立点の変更可能範囲が非常に狭いという弱点がある。

錯視立体の再モデリング処理とレンダリング処理を独立に動作させることは難しく、汎用レンダリング環境への移植は困難である。

以上の検討結果から、中津らの錯視立体表現モデル^[3]を、汎用レンダリング環境下で動作するように改良・拡張し、他の3DCGモデルと共存可能な錯視立体表現モデルを考案する。

表1 従来手法における錯視立体表現モデルの比較
Table 1 Comparison of conventional illusion representation models

手法	① 3D形状モデル		② 写実的なレンダリング機能				③ 汎用レンダリング環境	④ シンブルかつ自動化された錯視の仕組み	⑤ 不可能立体の印象のロバスト性	
	立体モデル	隠面消去	線画	陰影処理	TM*1	反射屈折				錯視の仕組み
中津ら	×	×	○	×	×	×	アフィン変換による錯視立体生成	○変換は自動化済。 ○対応付けも自動化済。	○非常に広い	○
Wuら	○	○	○	○	○	○	Thin-Plate Spline 曲面の変形	○変換は自動化済。 ×変換される面・稜線・頂点間の対応付けは手作業で設定。	△極狭い	-
篠原ら	○	○	(○)*2	○	○	○	奥行値	○奥行値逆ソートはシステム化済。 ×奥行値を逆ソートする閾値設定は視点ごとに手作業で設定。	-	-
吉川ら	○	○	(○)*2	○	(○)*2	×	逆ソート	-	△狭い	-

注：*1 TM = Texture Mapping, *2 (○) = 未実装だが原理的に適合・可能

第4章

一般的な3DCGモデルと共存可能な

錯視立体表現モデル

中津らが提案した錯視立体表現モデル^[3]を、単に**既存表現モデル**と呼び、本論文で提案する錯視立体表現モデルを**拡張表現モデル**と呼ぶ。

既存表現モデル、拡張表現モデルとも、代表的な不可能図形のひとつである、トーラス状不可能図形を描画(表現)対象とする。

既存表現モデルの利点は前章で述べた。一方、線画で描画されることのみを想定しており、面の処理が考慮されていない点が弱点である。そのため、後述する疑似接続部の稜線の接続処理などは個別処理として扱われ、汎用性に乏しい。

また、既存表現モデル^[3]で扱われている座標系・初期形状定義は、数式によって明確に定義されていないものが含まれている。そこで、4.1節で新たに立式に適した座標系を定義する。4.2節で既存表現モデル、拡張表現モデルに共通する初期形状、および錯視成立点変更アルゴリズムによる3D形状変形処理を、4.1節で新たに導入した座標系で定義し直す。4.3節では、拡張表現モデルにおいて新たに導入する奥行整合処理について述べる。

4.1 錯視立体表現モデルと座標系・投影法

トラス状不可能図形のための錯視立体表現モデルの定義について述べる。既存表現モデル、既存表現モデルと拡張表現モデルの両モデルとも、その錯視立体は、図10に示すように、正四角柱で構成される。錯視立体を構成する正四角柱を単に**角柱**と呼ぶ。図10に示すように、この各角柱に#1から#4まで番号を付与する。

投影図上で角柱#1と角柱#4が重なっているように投影されて、図6(a)のようにあたかも接続しているように見えるが、実際には接続していない部分を**疑似接続部**(図11)と呼ぶ。疑似接続部以外で、角柱同士が3D空間上でも投影図上でも、両方で幾何学的に接続している部分を**接続部**と呼ぶ。

式の導出を簡単にするため、本章では、各角柱の太さを0として立式する。つまり、各角柱の代わりに線分を置いたものとする。実装時には、角柱の太さと接続部および疑似接続部の重なりを考慮する必要があるが、重なり具合は視点位置によらない定数である。重なりを含めた式の扱いは、本論文では省略する。

両モデルを描画するときには、平行投影を用いる。両モデルのローカル座標は、右手系の直交座標系とし、原点は、X-Z平面に垂直に立っている角柱#1の上端(図10の点F)とする。

視点の位置座標は、経度 λ 、緯度 ϕ を用いて極座標系 (λ, ϕ) で表す。視線は常に極座標系の原点を向いているものとする。経度 λ はZ軸からX軸への偏角で、緯度 ϕ はZ軸からY軸への偏角である。平行投影であるため、極座標系の原点の取り方や動径については議論しない。また、スクリーンの大きさやカメラのクリッピング距離はここでは考えない。

既存表現モデルおよび拡張表現モデルでは、視点が極座標系の極の位置になると、角柱#1の長さが無限大に発散してしまう。また、実装時には、角柱の太さの分だけ、極からずれていても不可能図形として描画されない範囲ができる。そのため、錯視成立点となることができるのは、極座標の極付近を除いた範囲であり、これを視点の移動可能範囲とする。

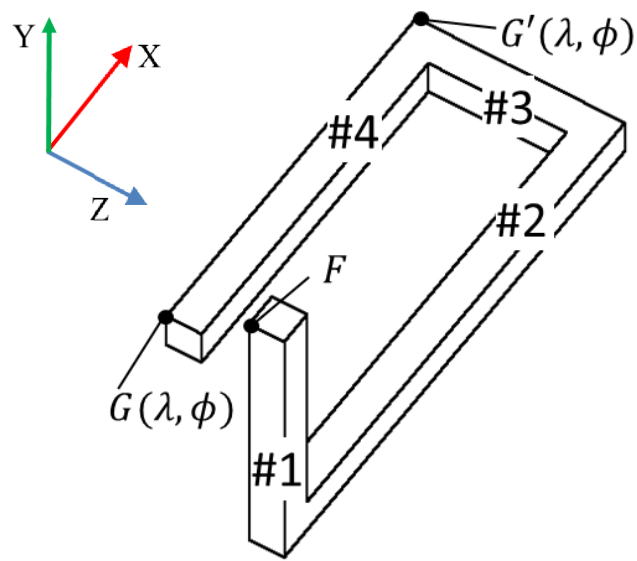


図10. 錯視立体表現モデルの3D形状

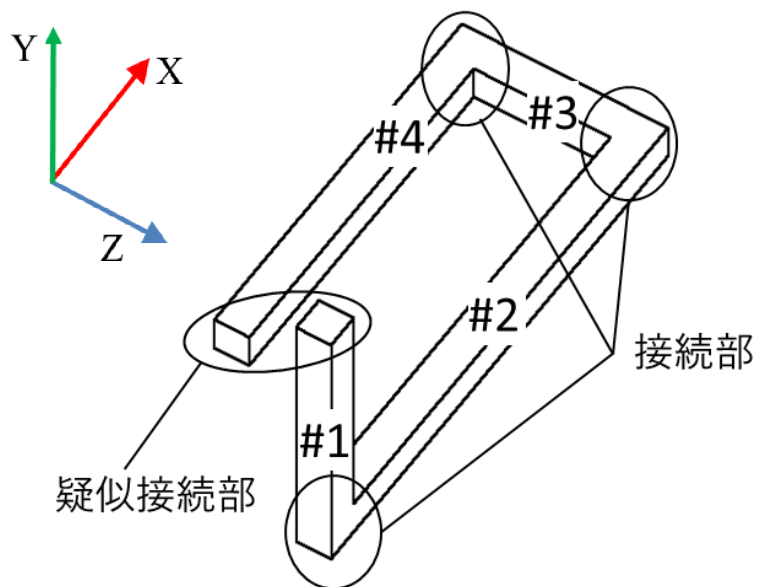


図11. 錯視立体の疑似接続部と接続部

4.2 錯視立体表現モデルの形状定義と再モデリング処理

既存表現モデルと拡張表現モデルに共通する初期形状の定義，および錯視視点の変更のための再モデリング処理について述べる．

本節で述べる初期形状は，文献^[3]で初期視点位置を適切に選んだ場合と同じ形状になる．

錯視成立点の変更に合わせて拡張表現モデルの3D形状を再モデリングする，錯視視点の変更処理は，既存表現モデルでも用いられた**3D形状変形処理**(再モデリング処理(1))と，その後続く疑似接続部に関わる遮蔽関係を修正する**奥行整合処理**(再モデリング処理(2))の2段階からなる．

本節では，第1段階である3D形状変形処理について述べる．3D形状変形処理の詳細や，個別の移動の効果に関しては，文献^[3]に詳しく記載されているので，ここでは最低限の記述に留める．第2段階の奥行整合処理の詳細は次節で述べる．

なお，以下の記述の中で，錯視立体表現モデルというときは，既存表現モデルと拡張表現モデルの両方を示すことがある．

4.2.1 錯視立体表現モデルの初期形状

錯視立体表現モデルを観察する視点位置の初期値を $(\lambda_0, \phi_0) = (0, \pi/4)$ とする。これを初期視点位置とよぶ。初期視点位置 (λ_0, ϕ_0) が錯視成立点となる錯視立体表現モデルの3D形状を、初期形状とよぶ。

錯視成立点が視点位置 (λ, ϕ) となるような錯視立体をなす角柱# N ($N = 1, 2, 3, 4$)の長さを $L_N(\lambda, \phi)$ と記す。

初期視点位置を錯視成立点とするためには、図12に示すように、角柱#1と角柱#3の長さ、角柱#2と角柱#4の長さがそれぞれ同じになればよい。すなわち、任意の正の定数を r, s とすると、式(1), (2)によって、初期形状の角柱の長さ $L_1(\lambda_0, \phi_0) \sim L_4(\lambda_0, \phi_0)$ が定まる。

実装時には、定数 r, s 、および角柱の太さを、描画したい不可能図形の形状に合わせて手動で定めればよい。

$$L_1\left(0, \frac{\pi}{4}\right) = L_3\left(0, \frac{\pi}{4}\right) = r \quad (1)$$

$$L_2\left(0, \frac{\pi}{4}\right) = L_4\left(0, \frac{\pi}{4}\right) = s \quad (2)$$

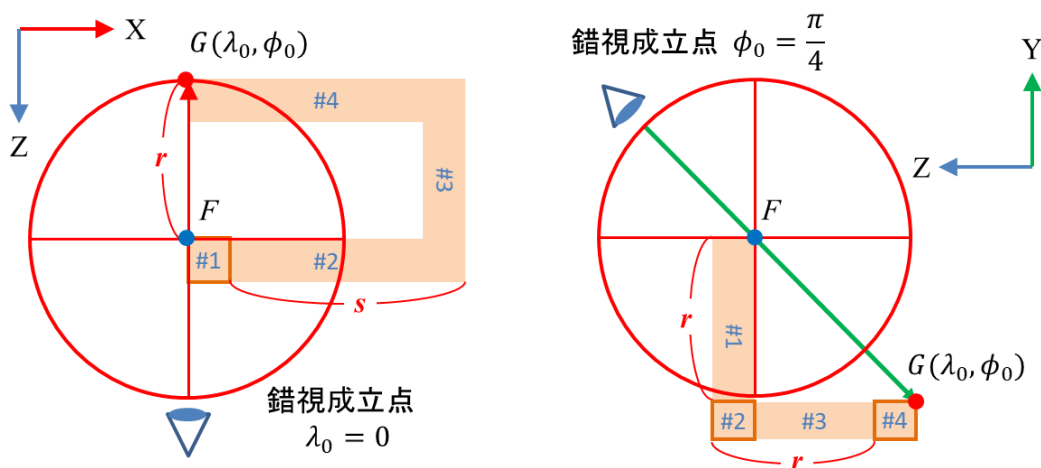


図12. 錯視立体表現モデルの初期形状

4.2.2 再モデリング処理(1)：3D形状変形処理

図8に示したように、それぞれの視点で不可能図形として描画されるように、錯視立体表現モデルの3D形状を動的に再モデリングする。

図13に示すように、視点が初期値 (λ_0, ϕ_0) から (λ, ϕ) に移動したとき、角柱#1の端点 F と角柱#4の端点 G が投影図上で同一の位置にあって、互いにつながっているように見せるために、角柱#4を移動・伸縮する。このとき、角柱#4の点 G の座標は式(3)で与えられる。

$$G(\lambda, \phi) = \begin{pmatrix} -r \sin \lambda \\ -r \tan \phi \\ -r \cos \lambda \end{pmatrix} \quad (3)$$

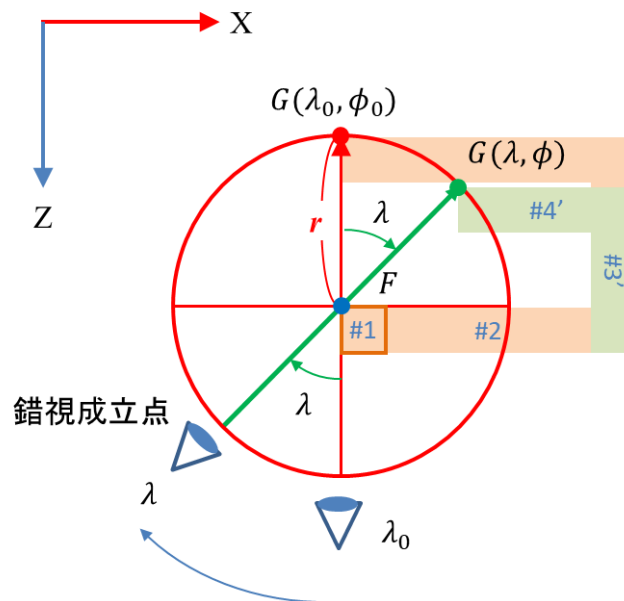


図13. 錯視立体表現モデルの点 G の移動

次に、角柱#3と角柱#4の接続部である点 G' について、式(4)で示される幾何変形を行う (図14).

$$G'(\lambda, \phi) = \begin{pmatrix} -\frac{r}{k} \sin \lambda + s \\ -r \tan \phi \\ -r \cos \lambda \end{pmatrix} \quad (4)$$

ここで、係数 k は、伸縮する度合いを調整する係数で、**伸縮制御係数**と呼ぶ。 k の値が0に近づくほど、大きく伸縮する。

中津ら^[3]によれば、初期形状の定義に用いられる r, s や太さに適した k を用いて再モデリングすることで、変形を自然に感じさせることができる。すなわち、式(4)の変形が印象のロバスト性向上に寄与している。

図14に示すように、角柱#3は角柱#4に、角柱#2は角柱#1に、それぞれ接続されるように、移動・伸縮する。

以上の手法により得られた既存表現モデルの描画結果が図8であり、拡張表現モデルでも同様の結果が得られている。

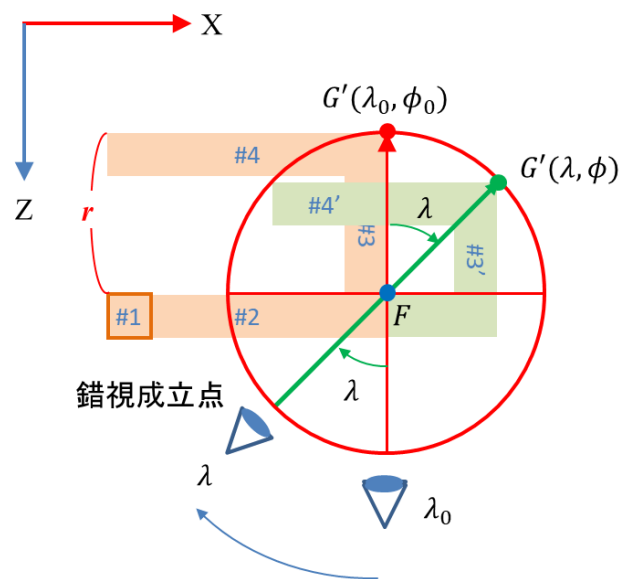


図14. 錯視立体表現モデルの点 G' の移動

4.3 錯視立体表現モデルの拡張

4.3.1 既存表現モデルで発生する問題

既存表現モデルから拡張表現モデルに移行するには、ワイヤーフレームモデルで定義された既存表現モデルに面情報を追加して単にサーフェスモデルに変換しただけでは、錯視視点の移動によって錯視立体の遮蔽関係が矛盾したまま描画され、不可能図形でなくなってしまう。

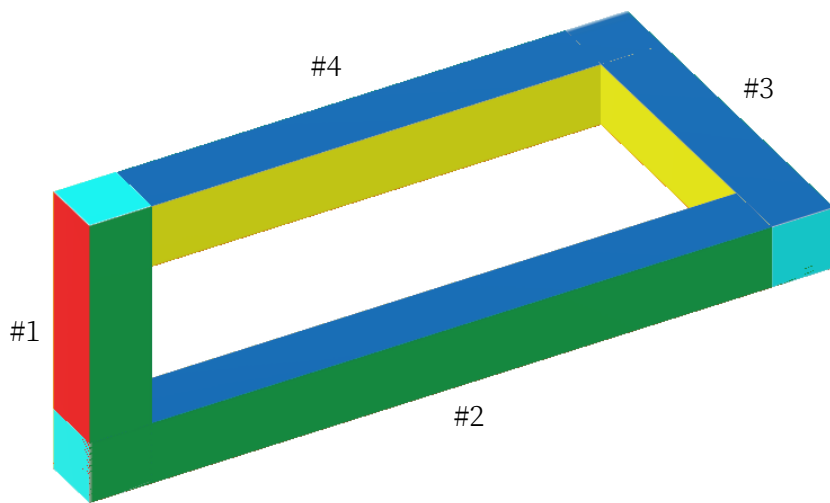
具体例として、4.2.2項で述べた既存表現モデルの3D形状変形処理の結果を図15に示す。

視点の経度 λ が $\lambda > 0$ のとき、図15(b) の破線部のように、遮蔽関係が矛盾して描画され、不可能図形でなくなってしまう。

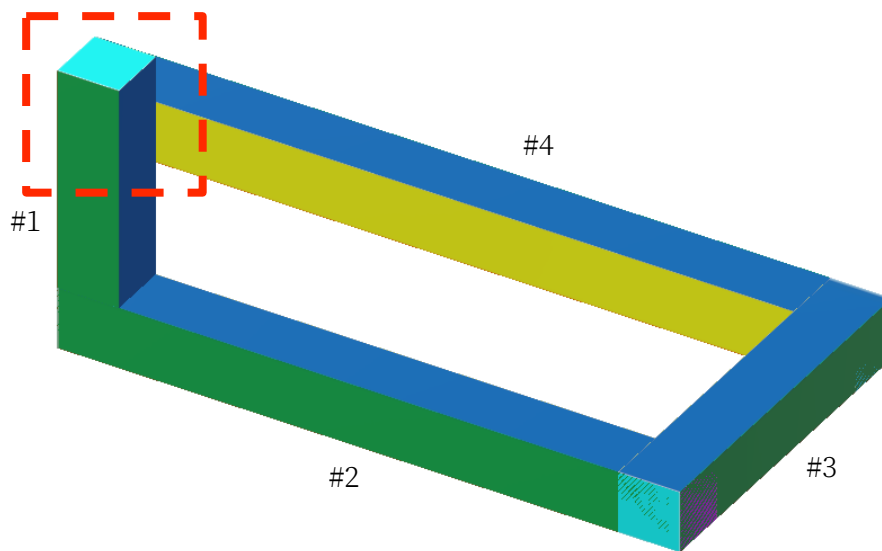
一方、視点の経度 λ が $\lambda < 0$ のときは、遮蔽関係の矛盾は生じない。

遮蔽関係の矛盾は、角柱#1の角柱#2側の面（濃い青色の面）が角柱#4の面（黄色と青の面）を遮蔽することにより生じる。既存表現モデルの3D形状変形処理では、角柱#1は角柱#4の左端より常に視点に近い位置となる。一方、投影図が不可能図形になるためには、角柱#4は角柱#1から手前に向かって伸びた形状として知覚される必要がある。これが矛盾の原因となる。

この問題を解決するために、再モデリング処理の第2段階として、次項で、奥行整合処理を提案する。



(a) ($\lambda < 0, \phi > 0$)



(b) ($\lambda > 0, \phi > 0$)

図15.既存表現モデルで面の遮蔽関係が矛盾して描画された例

4.3.2 再モデリング処理(2)：奥行整合処理

視点位置が $\lambda > 0$ であるとき、拡張表現モデルは、既存表現モデルと同様に角柱#2, #3, #4を伸縮および平行移動させた後、図16に示すように、角柱#1の先端側に角柱#4'が、続いて角柱#3'が、順に接続するように平行移動する。すなわち、疑似接続部が常に視点位置から見て手前側に来るように角柱の並び順を変更する。このときの角柱の移動・伸縮の様子を図17に示す。

錯視立体表現モデルは平行投影されるので、角柱の接続順が変更されても同じ形状が投影されるため、投影図の形状が不可能図形となることに不都合は生じない。4.2.2項で述べた3D形状変形処理後に、式(5)の量だけ角柱#3, #4を平行移動することに相当する。

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} L_4(\lambda, \phi) - L_2(\lambda, \phi) \\ L_1(\lambda, \phi) \\ L_3(\lambda, \phi) \end{pmatrix} \quad (5)$$

これにより、 $\lambda > 0$ の範囲では、疑似接続部は角柱#2, #3'の間となり、視点から見て手前側にある疑似接続部から2本の角柱が伸びる形になるため、錯視立体から知覚される奥行感が整合され、疑似接続部に関わる遮蔽関係を修正できる。

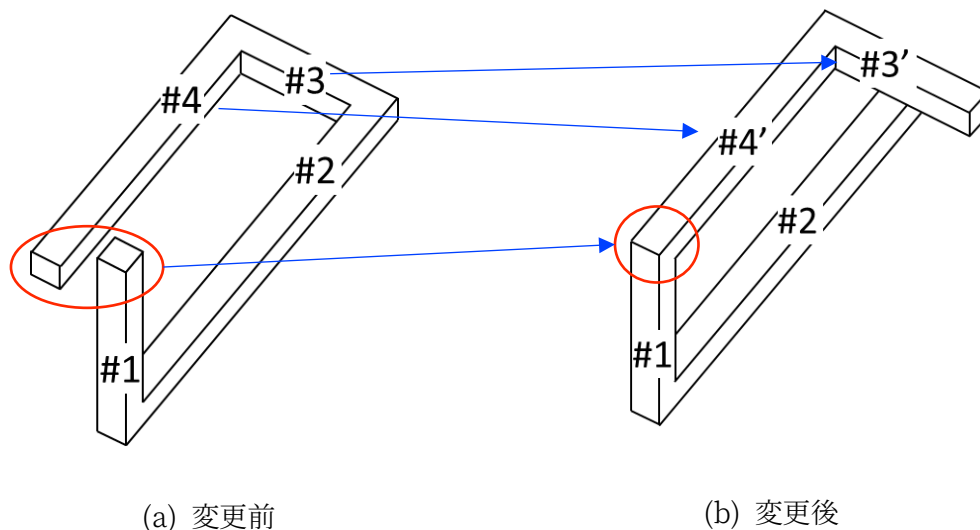


図16. 接続順を変更した錯視立体表現モデル

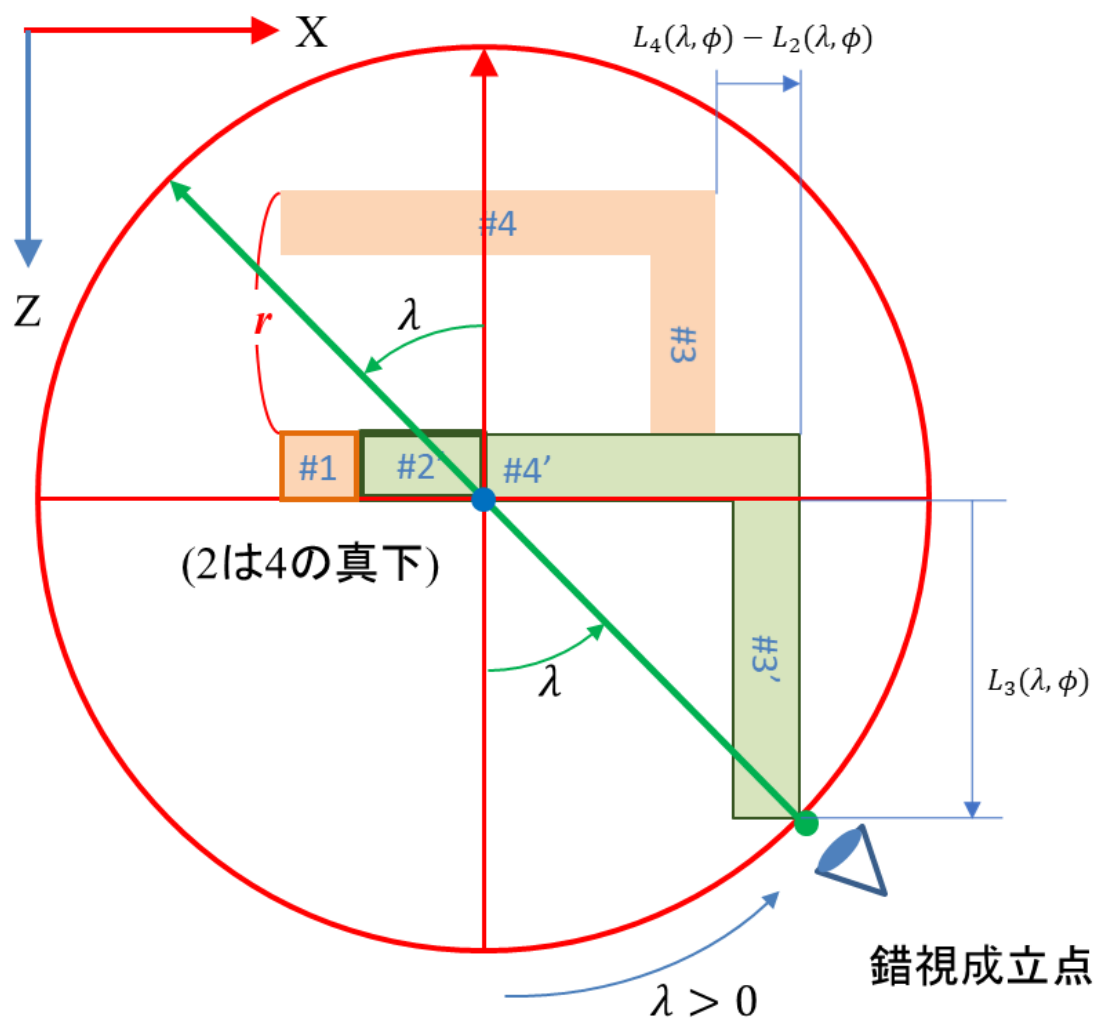


図17. 奥行整合処理による疑似接続部の変更

4.4 拡張表現モデルの実装と描画結果

本節では、提案した拡張表現モデルの汎用レンダリング環境への実装、拡張表現モデルを用いた不可能図形の描画結果を述べる。

4.4.1 拡張表現モデルを用いた不可能図形の描画結果

拡張表現モデルの再モデリング処理の第1段階、第2段階ともレンダリング・パイプラインから独立しているため、Unity^[11]上に拡張表現モデルを容易に実装できる(条件③)。サーフェスモデルを用いて(条件①)、写實的で豊富なレンダリング機能(条件②)を利用できる。

拡張表現モデルを用いた不可能図形の描画結果を図18に示す。図18は、視点の緯度 $\phi = \pi/6$ を固定し、経度 $\lambda \in [-\pi, \pi)$ ごとの変形結果である。図18に示すように、錯視立体の周りを視点 ϕ が1周回転しても、遮蔽関係の不整合が生じることなく、サーフェスモデルの不可能図形を描画することができた。

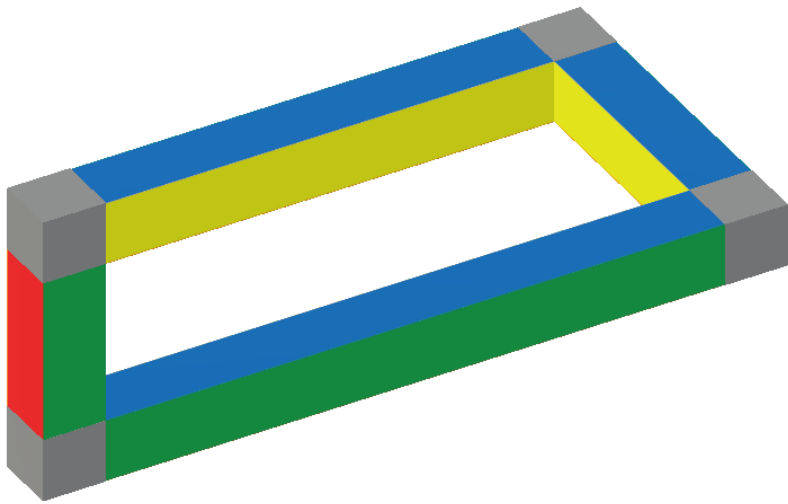
拡張表現モデルの描画結果である不可能図形(図18)と、既存表現モデルの描画結果(図19)は、同じ不可能立体の印象を想起させる。よって、拡張表現モデルは、シンプルかつ自動化された錯視の仕組み(条件④)、印象のロバスト性が高い(条件⑤)といった、既存表現モデルの利点を継承している。

すなわち、条件①～⑤を満たしたことから、拡張表現モデルは、他の3DCGモデルと共存可能であるといえる。

図15(b)で示した、サーフェスモデルに変換した既存表現モデルの描画結果と、拡張表現モデルの描画結果の比較を図20に示す。図20により、4.3.1項で挙げた問題点が改善していることが示された。

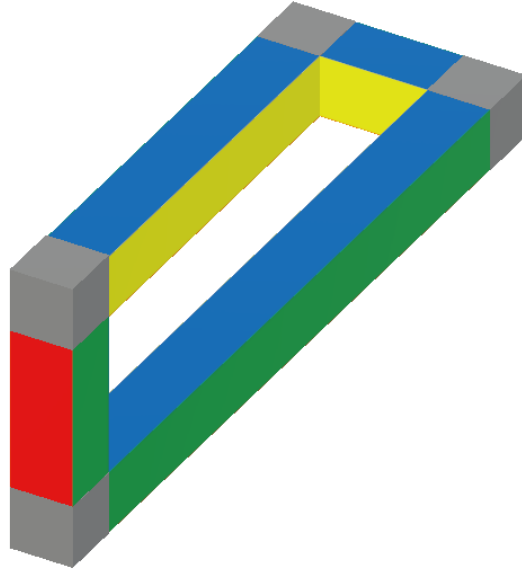


(a) $\lambda = 0$



(b) $\lambda = \frac{\pi}{6}$

図18. 提案した拡張表現モデルの描画結果

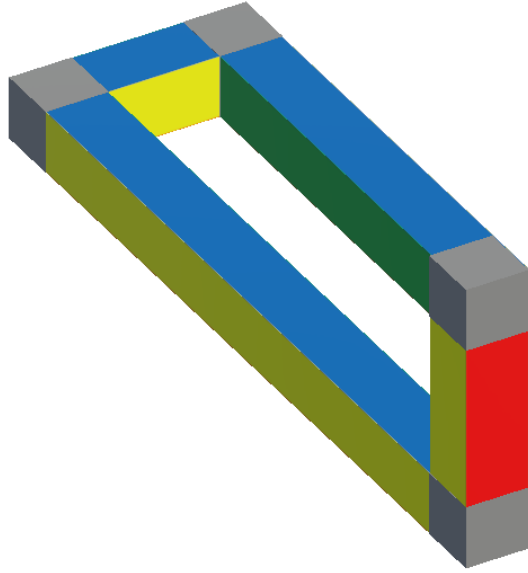


(c) $\lambda = \frac{\pi}{3}$

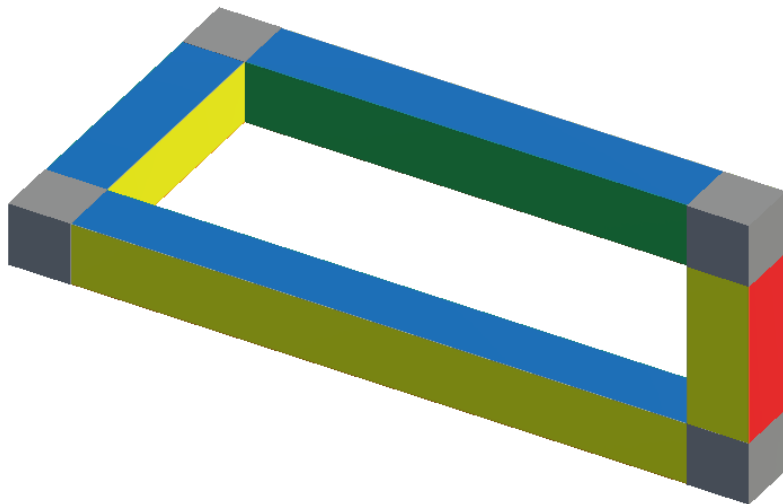


(d) $\lambda = \frac{\pi}{2}$

図18. 提案した拡張表現モデルの描画結果



(e) $\lambda = \frac{2\pi}{3}$

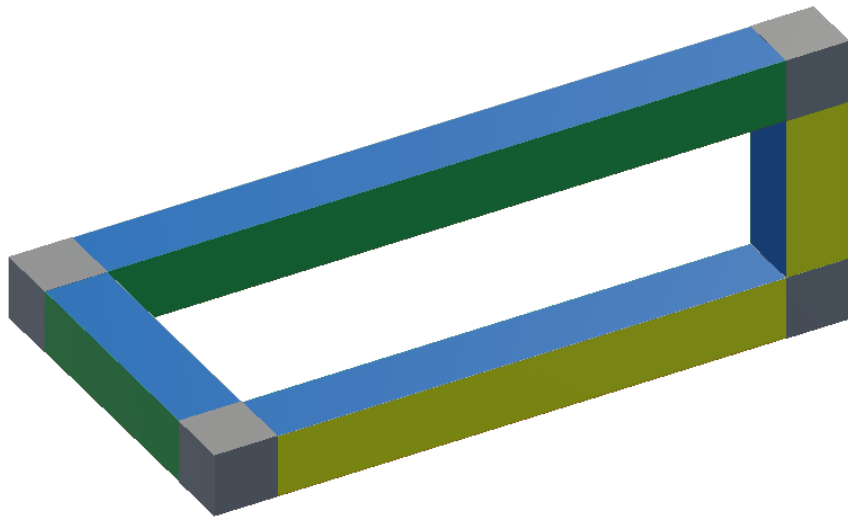


(f) $\lambda = \frac{5\pi}{6}$

図18. 提案した拡張表現モデルの描画結果

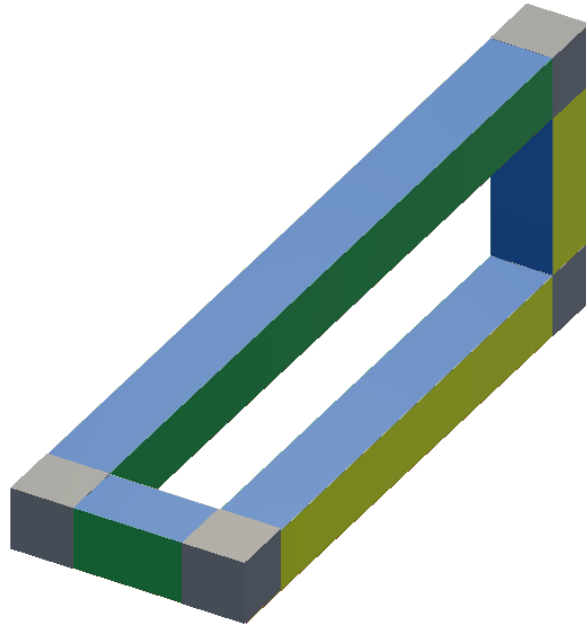


(g) $\lambda = -\pi$



(h) $\lambda = -\frac{5\pi}{6}$

図18. 提案した拡張表現モデルの描画結果

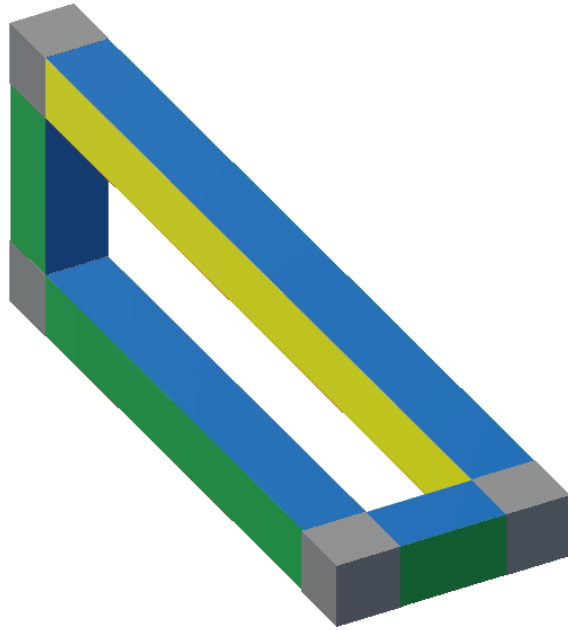


(i) $\lambda = -\frac{2\pi}{3}$

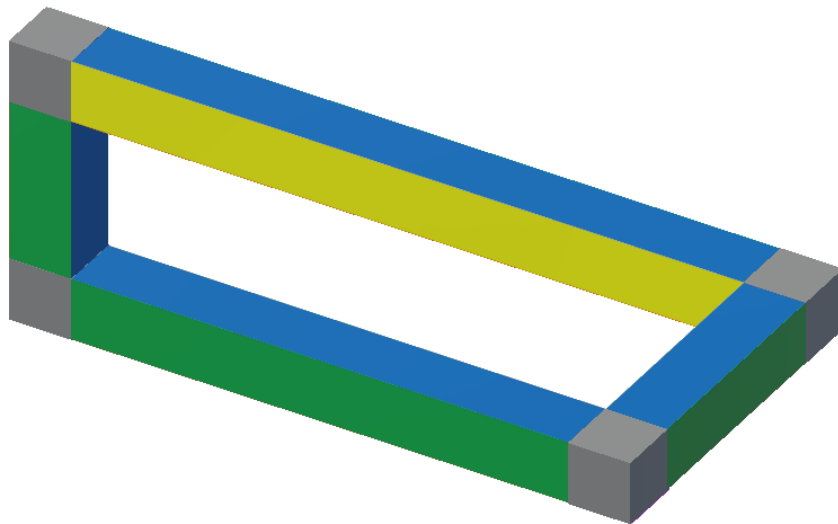


(j) $\lambda = -\frac{\pi}{2}$

図18. 提案した拡張表現モデルの描画結果



(k) $\lambda = -\frac{\pi}{3}$



(l) $\lambda = -\frac{\pi}{6}$

図18. 提案した拡張表現モデルの描画結果

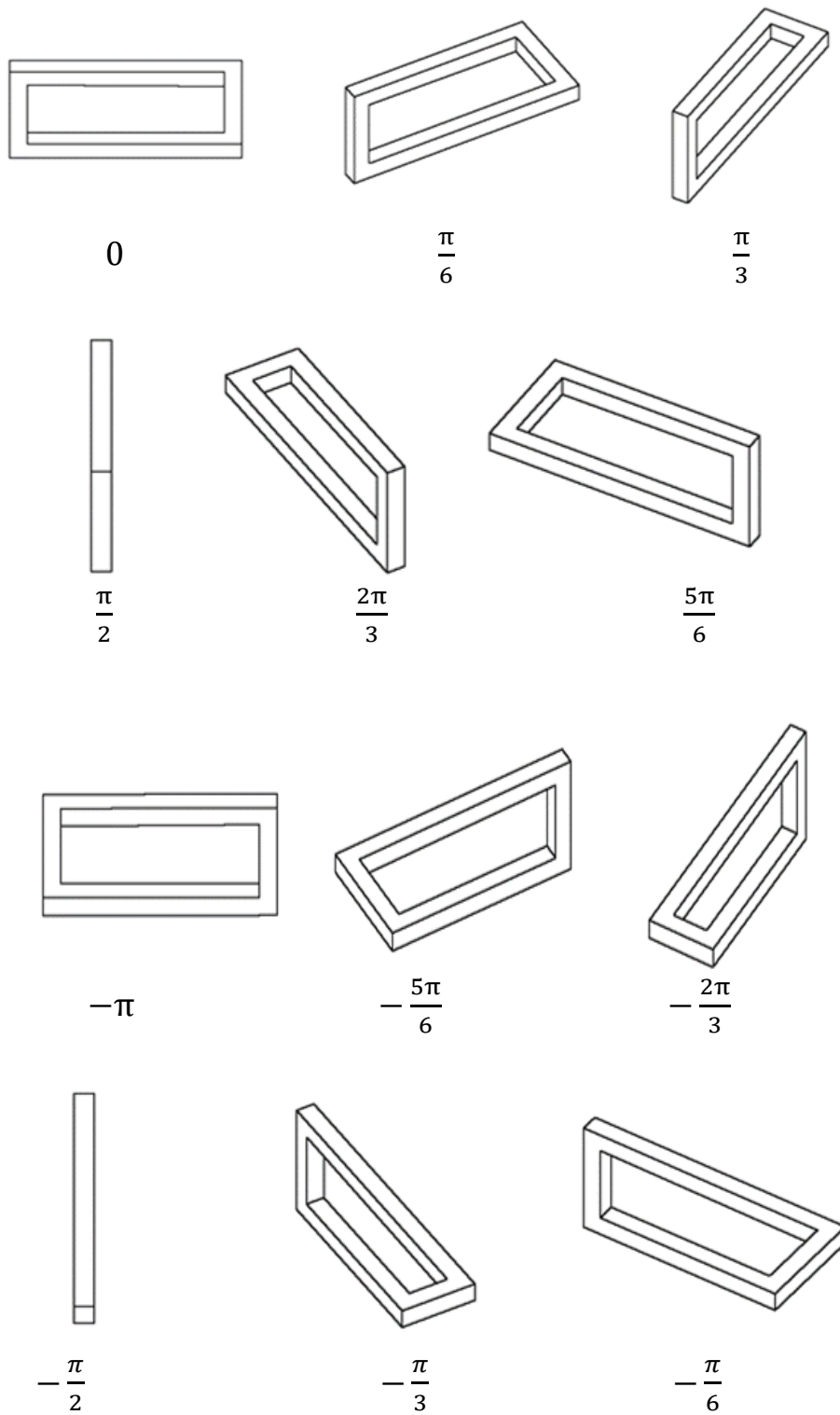
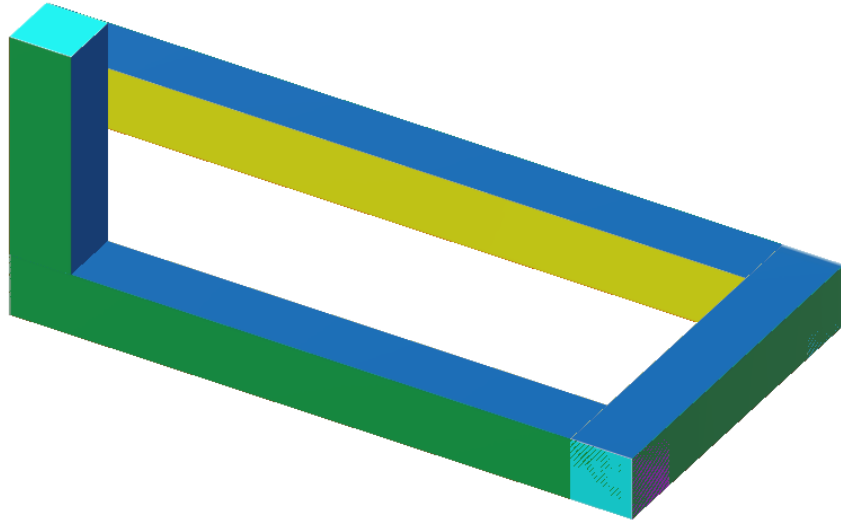
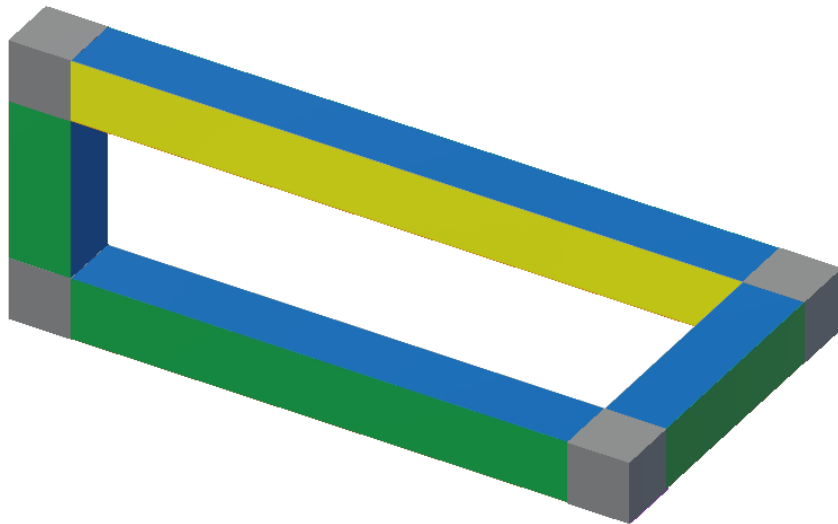


図19. 既存表現モデルの描画結果(図8の再掲)



(a) 既存表現モデルに面を貼った描画結果



(b) 拡張表現モデルの描画結果

図20. 既存表現モデルと拡張表現モデルの描画結果の比較 ($\lambda > 0, \phi > 0$)

4.4.2 汎用レンダリング環境下での実装

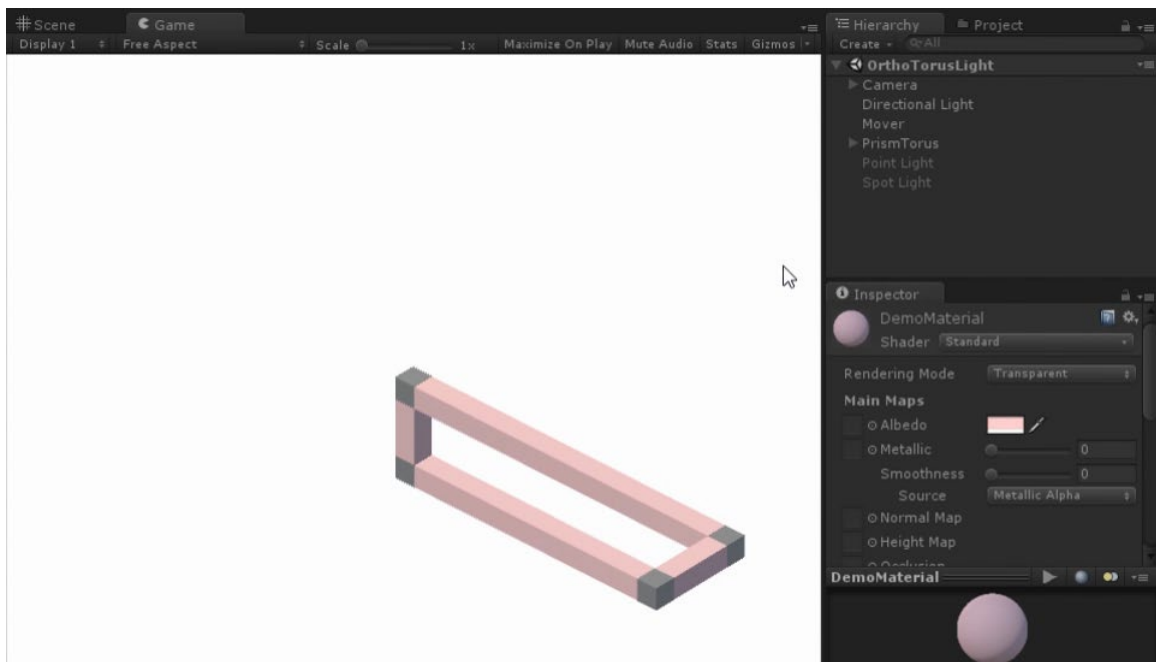
提案した拡張表現モデルを描画するシステムを, Unity^[11]上に実装した. 同システムの画面例を図21に示す.

この実装では, 既存手法^[3]と同様に視点の移動に伴ってリアルタイムで不可能図形を表示可能なだけでなく, Unityの機能により, 対話的に錯視立体の表面素材の変更操作を行うことができる.

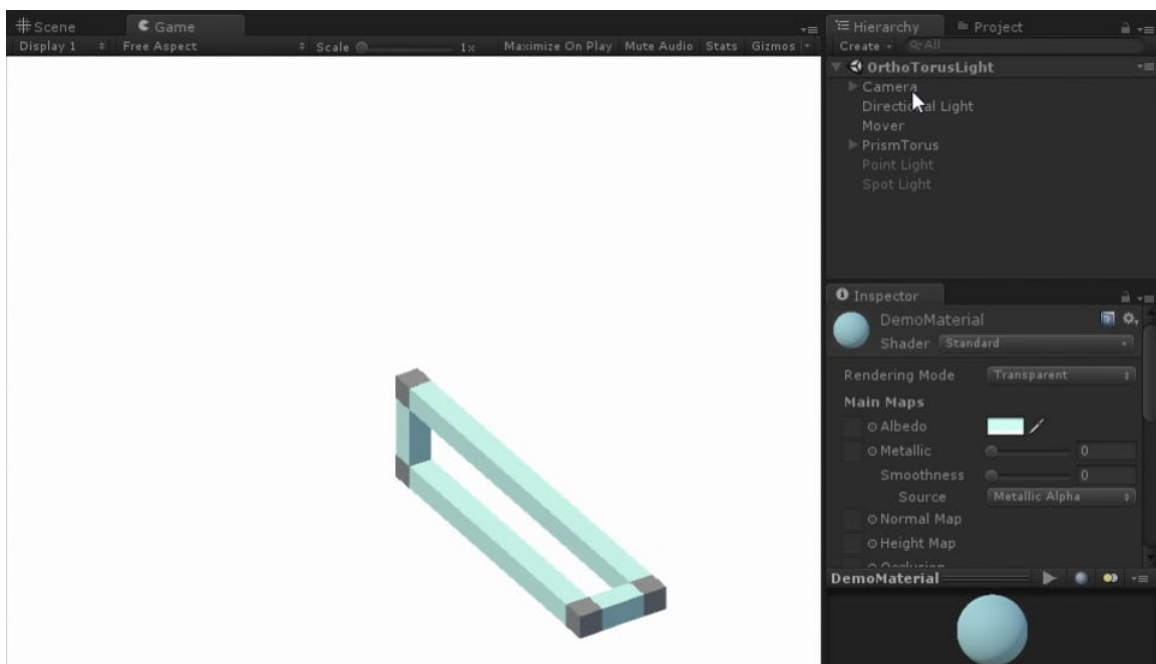
図21(a), (b)は, 表面色を変化させた例である. また, 図21(c)は, 表面素材を金属風にした例である. 図21(d)は, 透明素材を用いた表現を行った例である.

Unity上に実装することによって, Unityを用いる他のソフトウェア・ハードウェアとの連携も可能となる. Unity上でKinect^[14]と連携した例を図22に示す. Kinectは撮影した人物の位置をリアルタイムで検出できるデバイスであり, そのSDK (Software Development Kit) はUnity向けのライブラリとして提供されている.

図22はKinectにより観察者の位置を検出し, 観察者の方向から見た不可能図形を表示するものである. 図22(a)は左から観察した場合の表示であり, 図22(b)は正面から, 図22(c)は右からの表示である.

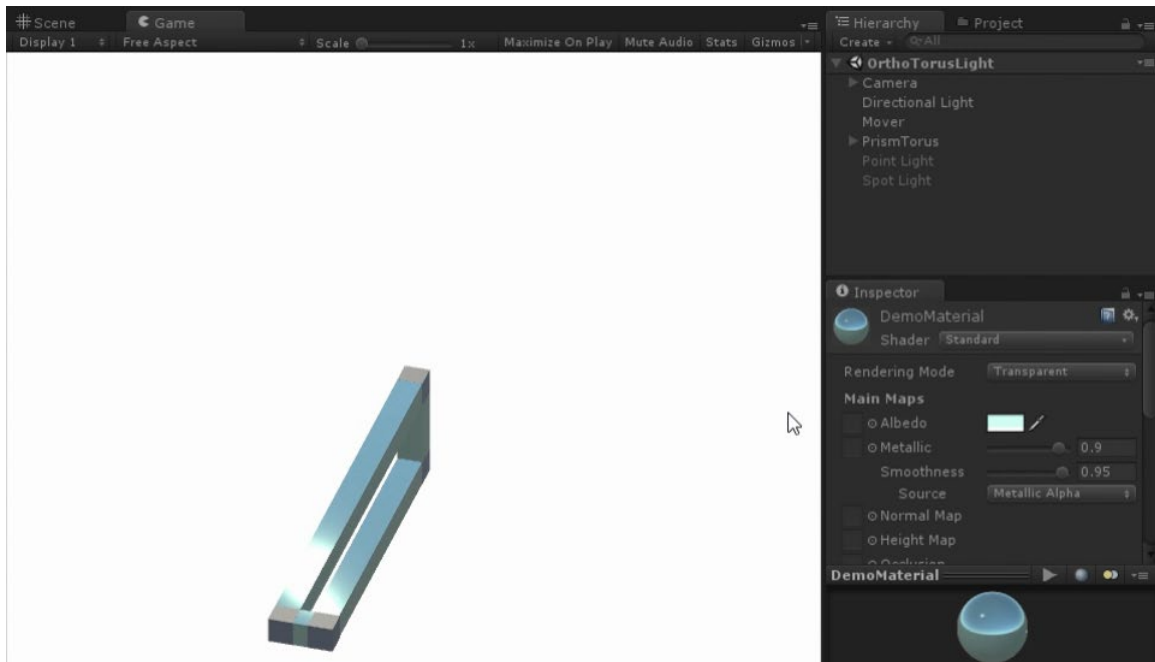


(a) 表面色1

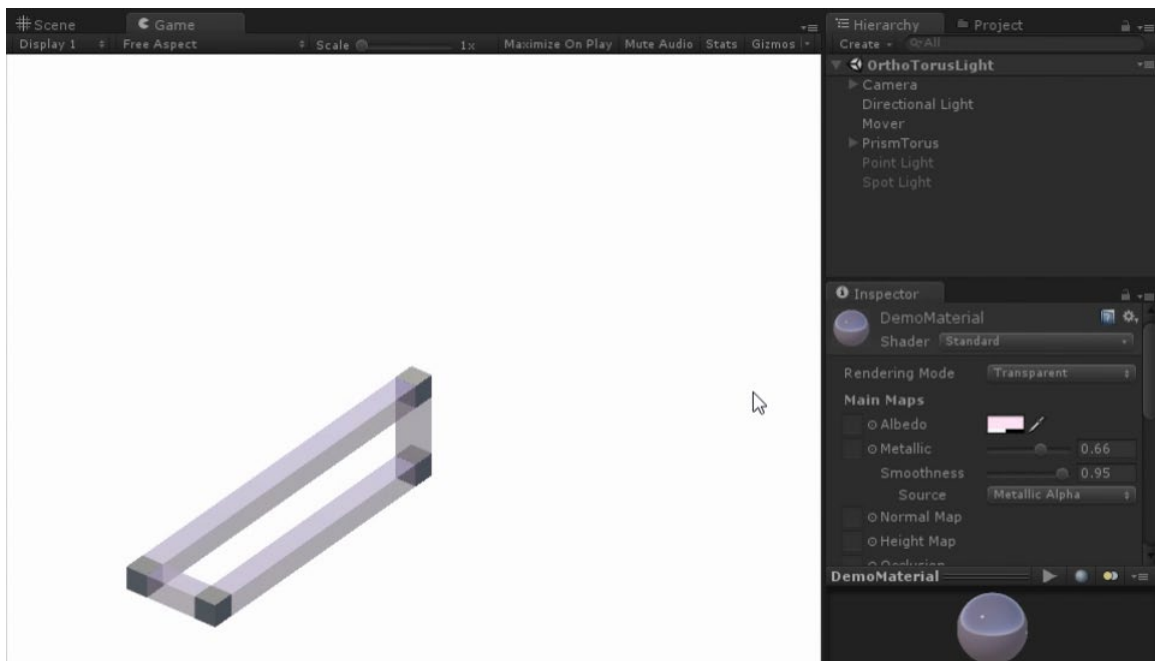


(b) 表面色2

図21. Unityによるリアルタイム動作画面



(c) 金属風表面



(d) 透明素材

図21. Unityによるリアルタイム動作画面



(a) 左からの観察時



(b) 正面からの観察時



(c) 右からの観察時

図22. Kinectによる観察者位置検出との連携

4.4.3 複雑な形状への応用・テクスチャマッピングの利用

拡張表現モデルは、同じ錯視の仕組みをもつ、より複雑な形状への応用や、錯視立体に対して、各種シェーダによるシェーディング処理や、高度なテクスチャマッピング処理を施すことが可能である。

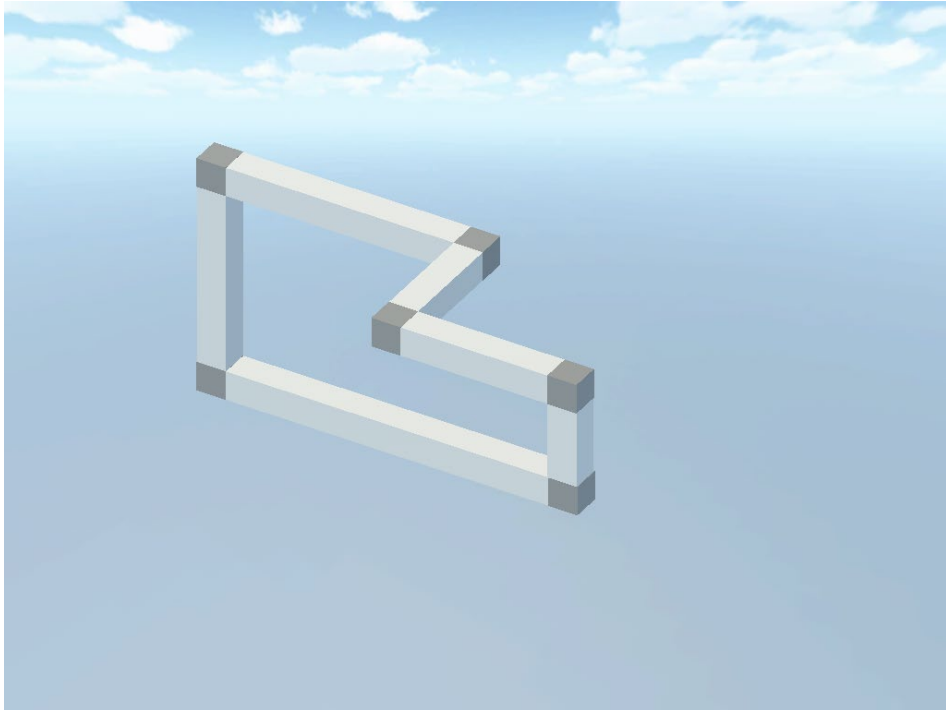
拡張表現モデルの形状の応用として、より複雑なトーラス状不可能図形を描画した。その結果を図23に示す。図23の錯視立体は、長さを約半分にした角柱3,4を、角柱1,2に平行な新しい角柱で接続したものである。図23に示すように、これまでより複雑な形状でも不可能図形として描画されている。

また、本手法のモデルにより面を描画できるようになったことで、図24に示すような、錯視立体をなす各角柱を円柱に置き換えた錯視立体モデルを利用することができるようになった。円柱によるトーラス状不可能図形は、稜線のみが描画される既存手法^[3]では表現することができず、面を描画できる錯視立体表現モデルでのみ描画が可能な表現のひとつである。

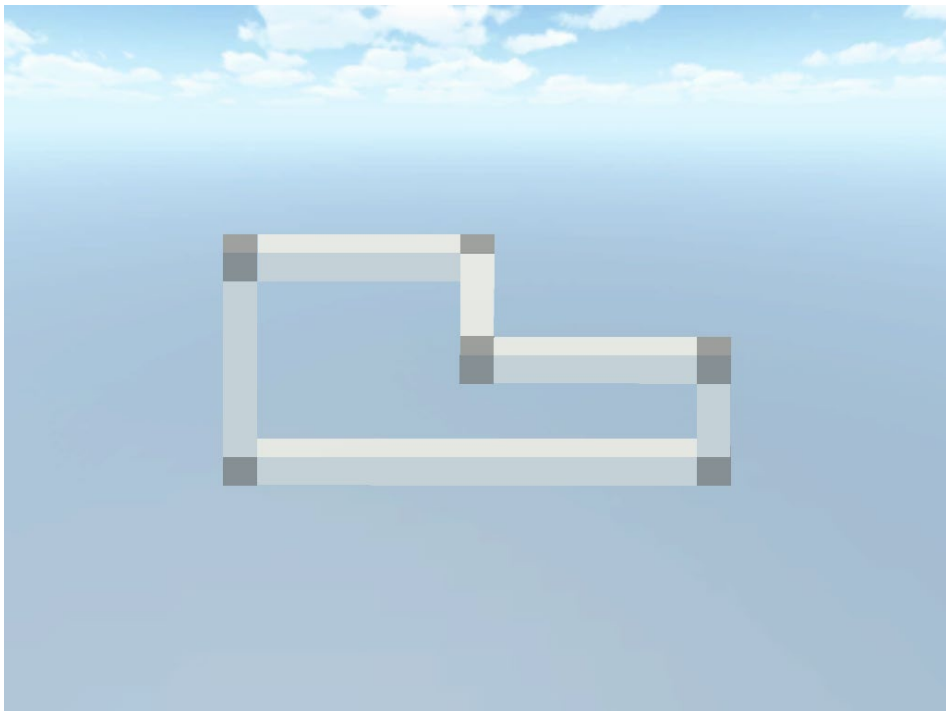
拡張表現モデルのサーフェスにテクスチャマッピングを施し、より複雑な形状を描画した結果を図25に示す。

錯視立体のサーフェスへのテクスチャマッピングは、Wuら^[4]と篠原ら^[9]も試みているが、錯視成立点を大きく動かすことができない。これに対して、拡張表現モデルは、図25に示すように、不可能立体としての印象がロバストな錯視立体モデルのサーフェスにテクスチャをマッピングできるので、その錯視立体表現モデルを含んだ3DCGシーンのカメラワークの自由度が高い。実装に用いたUnity^[11]には、マッピングされる位置をフレームごとに変える機能が備わっている。この機能を使って、図25に示すように、ビデオ・テクスチャをマッピングし、水の流れている様子を表現することができる。

なお、拡張表現モデルの各角柱は視点位置によって伸縮するため、テクスチャの接続には工夫を要する。図25の例では、角柱の接続部を分離して、接続部及び疑似接続部は静的なテクスチャ、角柱部分は時系列で動的に変化するテクスチャをマッピングすることで、テクスチャの不連続が現れないようにしている。

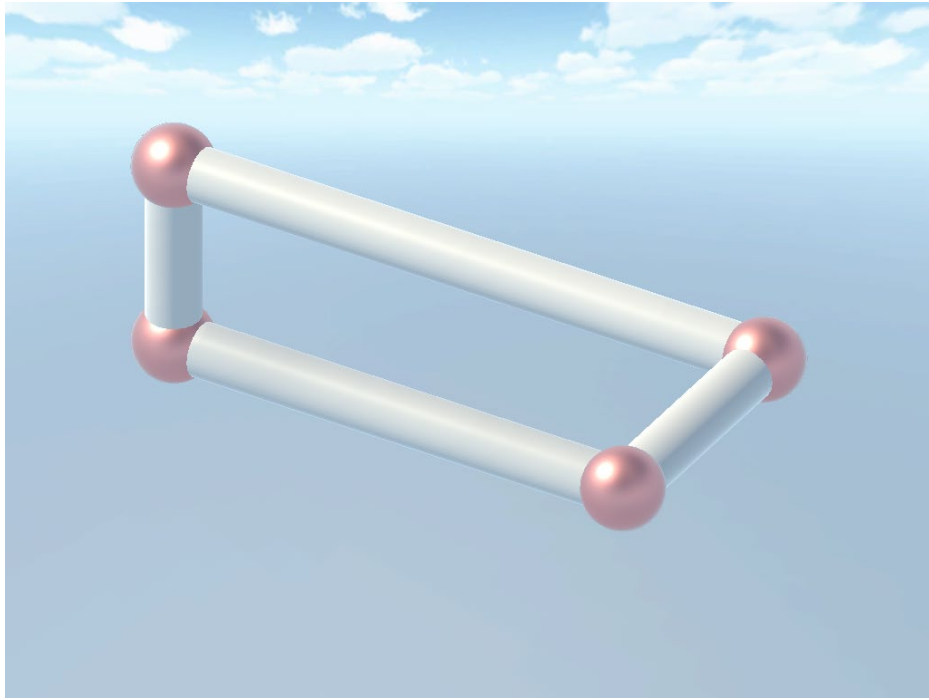


(a) 斜め

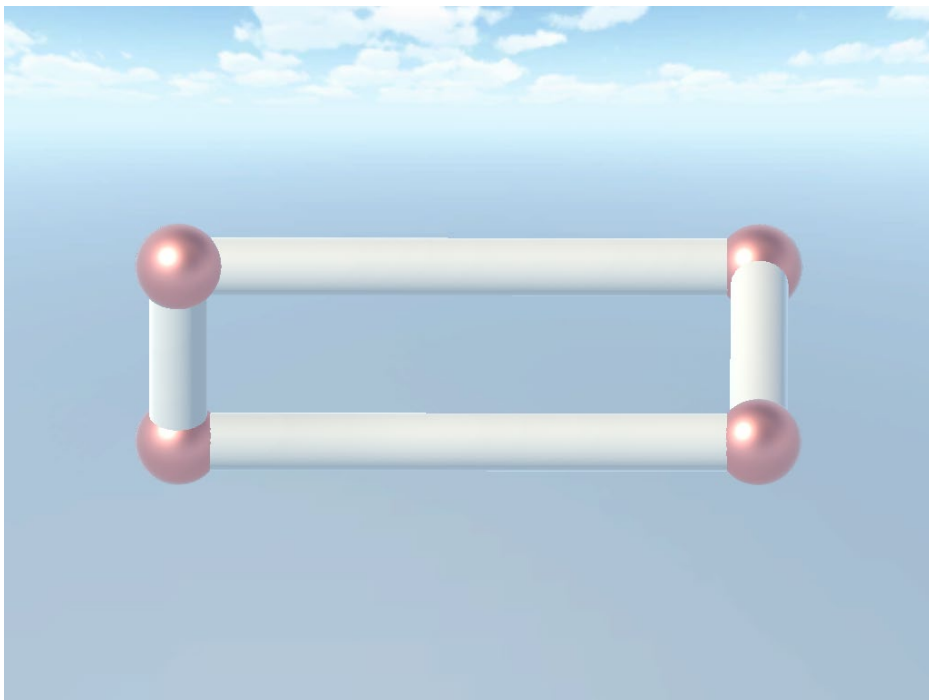


(b) 正面

図23. 凹みのあるトーラス状不可能図形の実出力結果



(a) 斜め

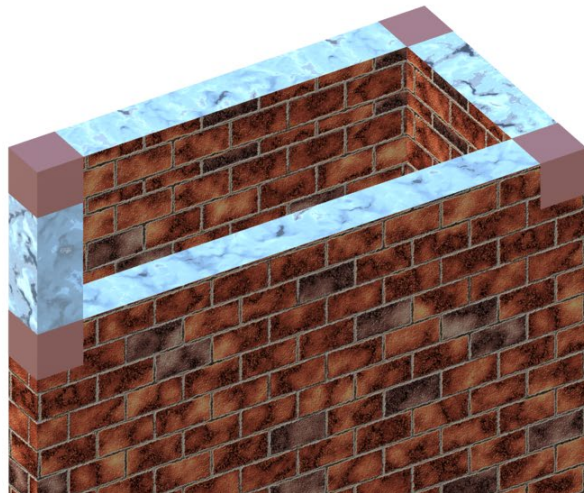


(b) 正面

図24. 円柱を用いたトーラス状不可能図形の出力結果

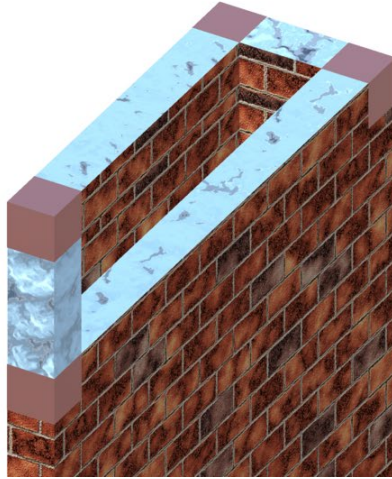


(a) $\lambda = 0$

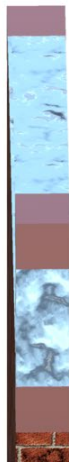


(b) $\lambda = \frac{\pi}{6}$

図25. テクスチャマッピングした不可能図形アニメーション

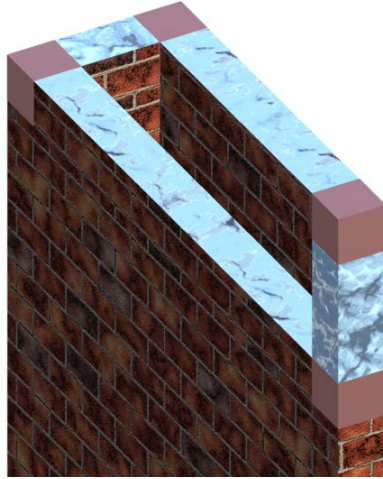


(c) $\lambda = \frac{\pi}{3}$

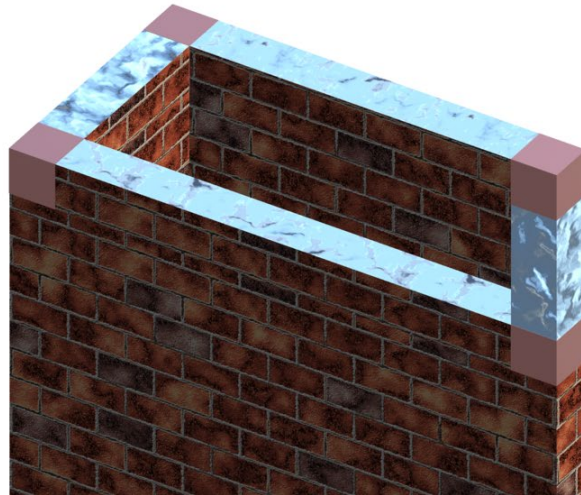


(d) $\lambda = \frac{\pi}{2}$

図25. テクスチャマッピングした不可能図形アニメーション



(e) $\lambda = \frac{2\pi}{3}$

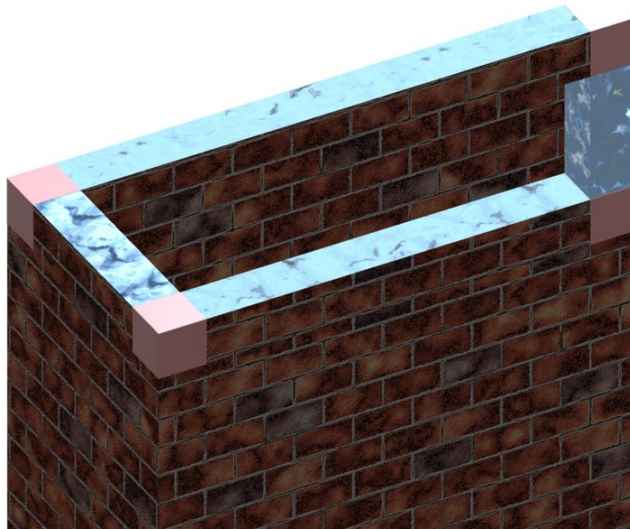


(f) $\lambda = \frac{5\pi}{6}$

図25. テクスチャマッピングした不可能図形アニメーション

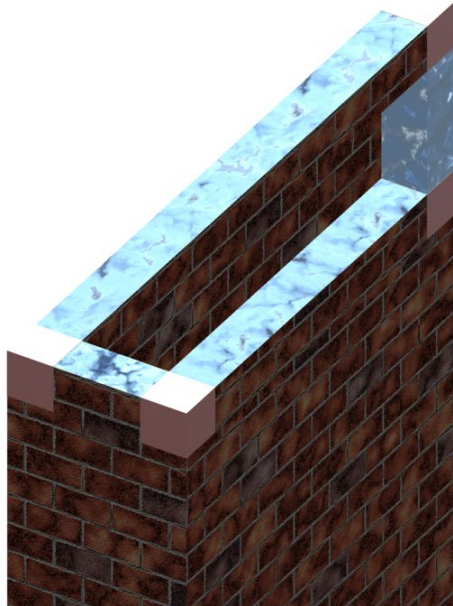


(g) $\lambda = -\pi$



(h) $\lambda = -\frac{5\pi}{6}$

図25. テクスチャマッピングした不可能図形アニメーション

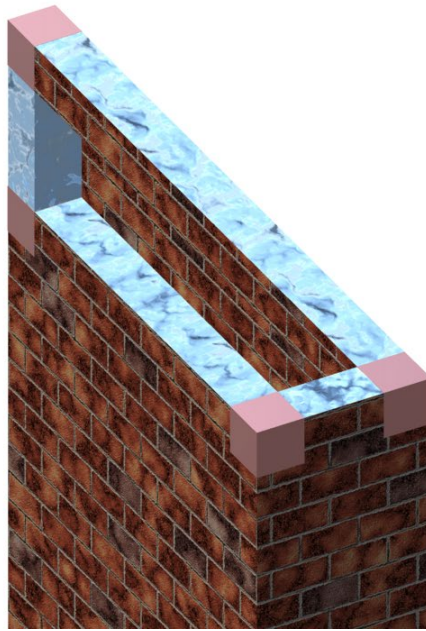


(i) $\lambda = -\frac{2\pi}{3}$

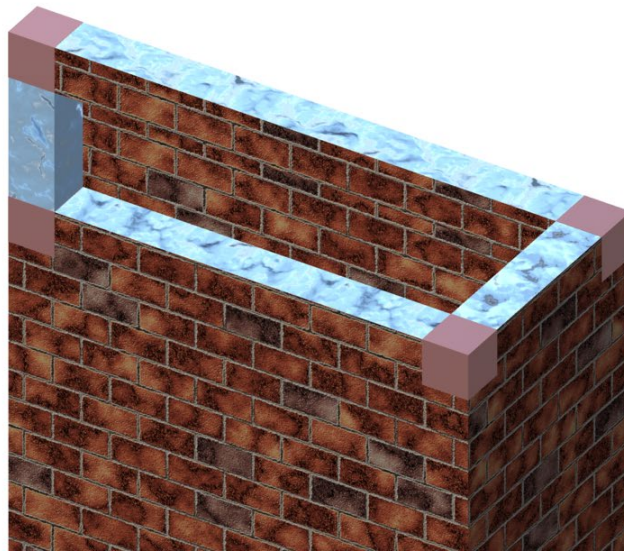


(j) $\lambda = -\frac{\pi}{2}$

図25. テクスチャマッピングした不可能図形アニメーション



(k) $\lambda = -\frac{\pi}{3}$



(l) $\lambda = -\frac{\pi}{6}$

図25. テクスチャマッピングした不可能図形アニメーション

第5章

不可能立体の付影処理

影の有無は写実的な画像生成のクオリティに大きく影響する。Unityをはじめとする汎用レンダリング環境では、付影処理が標準実装されているが、これは、一般的な3DCGモデルを対象としたものである。不可能立体は印象上には存在しない立体であり、不可能図形から想起される、印象としての不可能立体の影はどのような形であるべきか、今まで議論されて来なかった。篠原ら^[9]は影を含む不可能図形の描画結果を報告しているが、技術的詳細は述べられていない。

第1章で述べたように、不可能図形のトリックアートとしての面白さは、不可能立体をイメージできるが、現実には存在し得ないという矛盾や違和感から生まれている。本章の目的は、不可能図形の影を描画することによって、観察者にとってトリックアートの面白さのもととなっている違和感を損なうことなく、不可能図形の面白さが増す印象を与えることである。そのために、「不可能立体の影」と知覚されるような影の形、すなわち、不可能立体の影の在り方を考察する。考察結果から、錯視立体を用いた影の描画方法を提案する。なお、本章では、光を遮り、影を発生させる原因となっている物体を、影の本体とよぶ。

5.1 不可能立体の影の在り方と錯視立体の影の要件

本節では、不可能立体の影の形状の在り方を考察する。

通常、物体の影は、観察者の移動によってその形状は変化しない。光源が移動すると、物体の影は常にその形状を変化させるのが自然な現象である。

一方、図26は、前章で提案した拡張表現モデルを用いて、カメラ視点を錯視成立点としている錯視立体の影を、Unity^[1]の標準機能により描画した結果である。このとき、影の形状の決定には、錯視立体を光源の位置から、つまり錯視不成立点から見た投影図が使われるので、錯視効果が失われている状態になる。その結果、図26のように不可能立体から得られる印象とは異なる影が描画されてしまう。影の形から錯視立体の仕組みが推測できてしまい、トリックアートとしての面白さを損なってしまう。また、観察者が影を錯視立体の影と理解できない恐れもある。

錯視立体の幾何的な3D形状と、観察者が想起した不可能立体の印象の形状は異なっており、その結果、図26のような影が描画されてしまう。つまり、不可能立体の影は、錯視立体の幾何形状そのものの影ではなく、印象の差異を吸収する影であることが求められる。付影手法により印象の差異が吸収されれば、不可能立体の影は、不可能立体を本体とする影であると観察者が感じられるようになる。

以上の考察結果から、錯視立体の影には以下の3つの要件を満たすことが求められる。

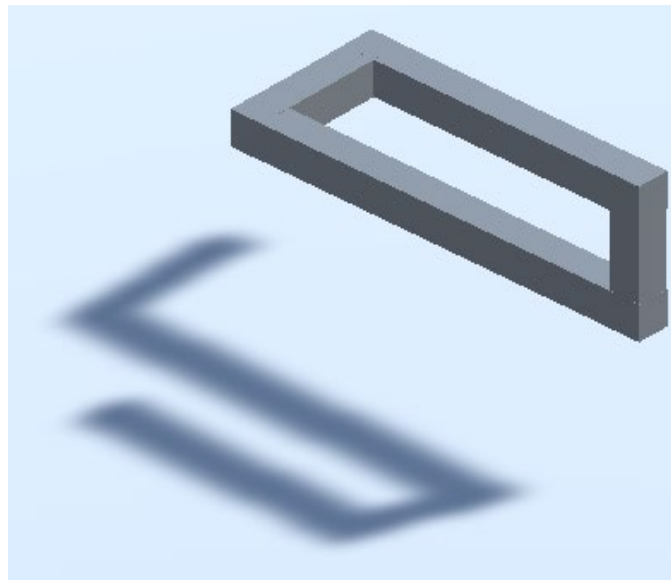


図26. 不可能立体の印象とは異なる影

- ① 光源が静止したまま，観察者の視点が移動しても影の形状が変化しない.
- ② 光源が移動すると，影の形状が変化する.
- ③ 印象の差異を吸収し，錯視立体からイメージされる不可能立体の影だと感じられる.

錯視立体の影の要件①②は，現実の影では自明なものであるが，不可能立体の印象を損なわないために，錯視立体の影にも同じ性質を持たせる必要がある.

5.2 Shadow Illusion Objects法の提案

一般に、3DCGで描画される影の形を、本体の形状とは別にコントロールするには、3DCGモデルの影を描画しないように設定した上で、Shadow Objects^[15]とよばれる、影描画専用の立体形状モデルを用いることが多い。この方法の利点は、付影アルゴリズムによらず実装可能なことである。レンダリング・パイプラインから見れば、「可視だが影を生成しないポリゴン」と「不可視だが影を生成するポリゴン」が存在するだけであり、ポリゴン数こそShadow Objectsの分だけ増加するものの、レンダリング・パイプラインを乱すことはない。

そこで、本論文では、錯視立体用のShadow Objectsとして、錯視立体の影の要件①～③を満たすShadow Illusion Objectsを提案する。Shadow Illusion Objectsは、以下の(i)～(iii)で定義される。

- (i) Shadow Illusion Objectsは、影の本体となる錯視立体表現モデルと同じ3D形状をもつ。
- (ii) Shadow Illusion Objectsは、錯視立体表現モデルと同じアルゴリズムで再モデリングされるが、カメラ視点ではなく、光源が錯視成立点となるように再モデリングされる。
- (iii) Shadow Illusion Objectsは、Shadow Objectsでもあるため、最終的な描画結果としてレンダリングされない (不可視である)。

ここで、影の本体となる錯視立体とは、可視である錯視立体のことであり、Shadow Illusion Objectsのことではない。

Shadow Illusion Objectsの定義(i)から、影は、その影の本体となる錯視立体が取り得る形から落ちた影となる。

Shadow Illusion Objectsの定義(ii)から、本体となる錯視立体が観察時にとっての形状とは異なる3D形状から落ちた影となる。同じく定義(ii)から、観察者が移動しても、光源が移動しなければShadow Illusion Objectsの形状も変化しない。すなわち、影の形状も変化しない (錯視立体の影の要件①適合)。一方、光源が移動すれば、同じく定義(ii)から、Shadow Illusion Objectsは光源が錯視成立点となるように再モデリングされ、影の形状が変化する (錯視立体の影の要件②適合)。

具体的な実装と描画結果は次節で説明するが、光源を錯視成立点としているので、図14のように錯視立体の仕組みが推測できてしまうような影を生成することもない。

なお、定義(ii)より、Shadow Illusion Objectsは、光源1つに対して、再モデリングされる。光源が複数ある場合には、光源の数だけ別のShadow Illusion Objectsを配置する。それによって、シーン内には複数の光源を置いた場合でも、影を生成することができる。ただし、こ

ここでは錯視立体表現モデルが平行投影のみに対応しているため、光源も平行光源のみに限定している。

Shadow Illusion Objectsの定義(iii)の通り、Shadow Illusion Objectsは、影を描画するが不可視と設定される。影の本体となる錯視立体表現モデルの3D形状は、影を描画しない設定とする。以上の定義と設定からなるShadow Illusion Objectsと錯視立体表現モデルを、一般的な付影アルゴリズムを実装したレンダリング環境でレンダリングすることで、不可能図形と影を合わせた画像を描画できる。

ここで、一般的な付影アルゴリズムとは、例えばZ Buffer法を用いた2段階法^[16]や、Ray Tracingによる付影法^[16]である。Shadow Objectsの仕組みは付影アルゴリズムに依存しないため、付影アルゴリズムの違いがShadow Illusion Objectsの定義に影響を及ぼすことはない。

Shadow Illusion Objectsを用いて描画した影は、本体となる錯視立体が取り得る形状から落ちた影ではあるが、その影が不可能立体を本体とする影として知覚されるかどうか(錯視立体の影の要件③)は観察者の主観によるところが大きい。後述の主観評価実験により、その妥当性を検討し、要件③を満たすことを示す。

5.3 Shadow Illusion Objectsを用いた付影処理の実装

前節で述べたShadow Illusion Objectsを実装した。実装には、Unity^[11]と、前章で提案した錯視立体表現モデル(拡張表現モデル)を用いた。付影処理のためのレンダリング手順は以下の通りである。

《初期設定》

- (1) シーン内に、錯視立体表現モデルの本体と同じ3D形状と、再モデリングアルゴリズムをもつShadow Illusion Objectsを配置する。
- (2) 錯視立体表現モデルのMesh RendererコンポーネントのGUIから、Cast ShadowsをOffに設定する。
- (3) Shadow Illusion ObjectsのMesh RendererコンポーネントのGUIから、Cast ShadowsをShadows Onlyに設定する。

《レンダリング・ループ》

- (4) Unityのカメラ視点と光源の位置を更新する。
- (5) 錯視立体表現モデルの再モデリング処理によって、カメラ視点が錯視成立点になるように、錯視立体表現モデルが自動的に再モデリングされる。
- (6) 同様に、光源位置が錯視成立点になるように、Shadow Illusion Objectsが自動的に再モデリングされる。
- (7) Unityでレンダリングを行う。

Unityに用意されたShadow Mapping法^[17]により、影が描画される。手順(2)(3)の設定に従って、錯視立体表現モデルの影は描画されず、Shadow Illusion ObjectsはShadow Objectsとして扱われる。

なお、本手順では触れなかったが、Cast ShadowsがOffになっているオブジェクトやShadows Onlyになっているオブジェクトが存在しても、レンダリング結果に問題は生じない。

- (8) 毎フレーム、以上の手順(4)～(7)を繰り返す。

錯視立体の影を含む3DCGシーンの描画結果を図27に示す。この描画結果から、提案したShadow Illusion Objectsを用いた影は、実際に、錯視立体の影の要件①②を満たすことが示された。

Shadow Illusion Objectsを用いて描画した影が不可能立体を本体とする影として知覚されるかどうか（錯視立体の影の要件③）を検証する主観評価実験については，次節で述べる．

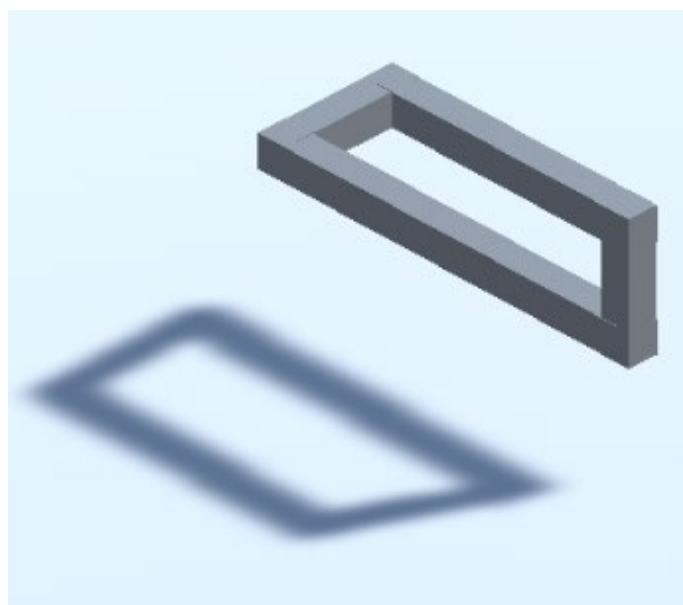


図27. Shadow Illusion Objectsによる錯視立体の影

5.4 付影処理の主観評価実験

前節のShadow Illusion Objectsを用いて生成した影が観察者にどのように見えているか、アンケートによる主観評価実験を行い、提案手法の妥当性を検証する。

5.4.1 主観評価実験環境

前章で述べたように、実験環境の実装にはUnity^[11]を用いた。本体の錯視立体、およびShadow Illusion Objectsの形状には、前章で提案した錯視立体表現モデル(拡張表現モデル)を用いた。

5.4.2 被験者と実験手順・条件

被験者は成人男女25名である。被験者には、まず、図1とともに、不可能図形とは何かを説明し、それが錯視によって不可能立体に見えることを理解してもらった。ただし、錯視立体の形状については説明しなかった。

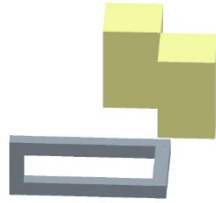
その後、被験者には3本の動画a, b, c (図28 (a), (b), (c))を順に視聴し、各動画に関する質問に回答してもらった。動画a, b, cの内容は以下のとおりである。

動画a. 錯視立体の影を描画しない動画 (図28(a))

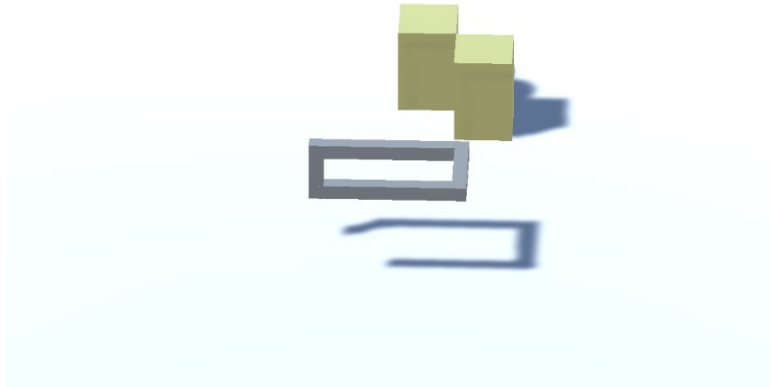
動画b. 描画されている錯視立体そのものの影を描画した動画 (図28 (b))

動画c. 提案したShadow Illusion Objectsを用いて影を描画した動画 (図28 (c))

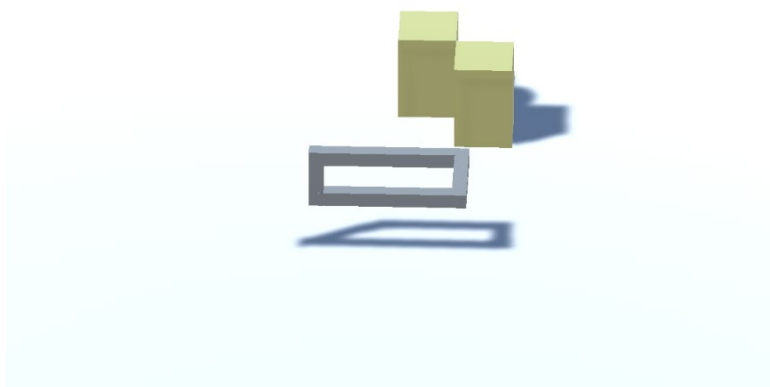
この実験は、影の形と不可能立体の印象との関係を分析することが目的である。そのため、視聴環境は特に指定しなかった。



(a-1) 評価用動画a：影なし， $\lambda = 0$

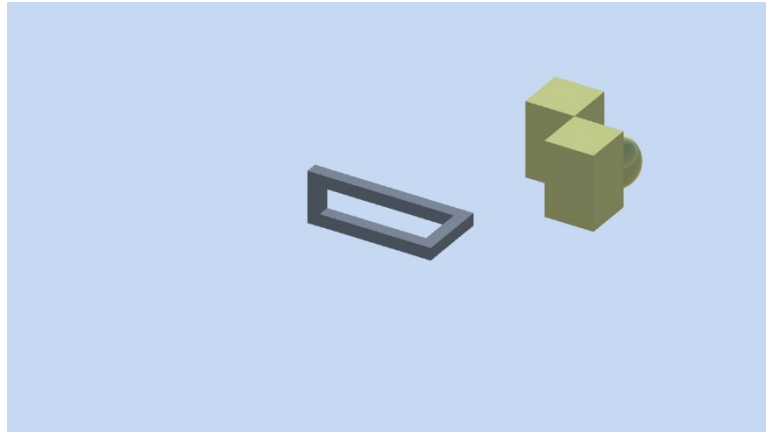


(b-1) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = 0$

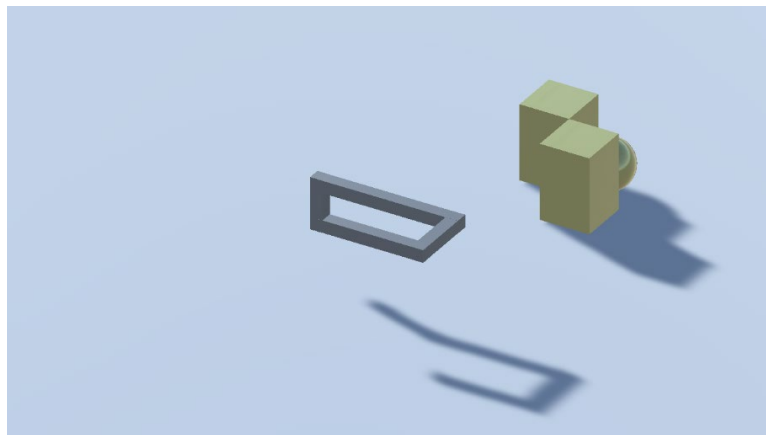


(c-1) 評価用動画c：提案手法による錯視立体と影， $\lambda = 0$

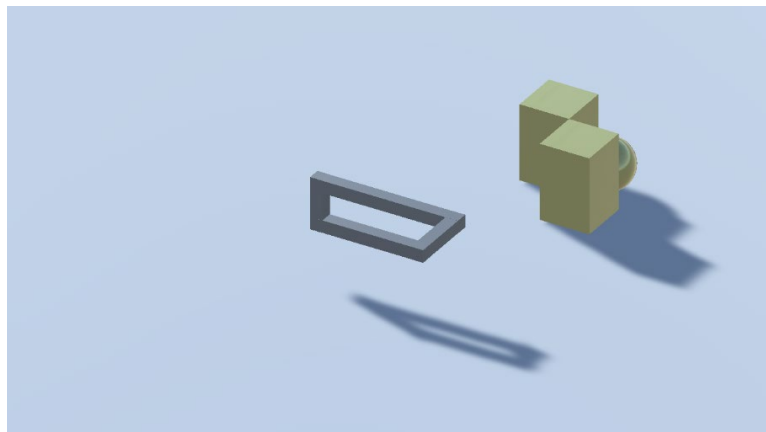
図28. 付影手法の主観評価実験に用いた動画



(a-2) 評価用動画a：影なし， $\lambda = -\frac{\pi}{6}$

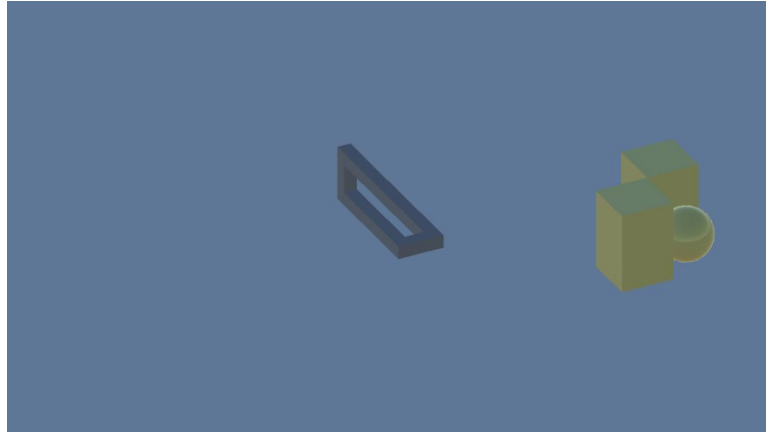


(b-2) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = -\frac{\pi}{6}$

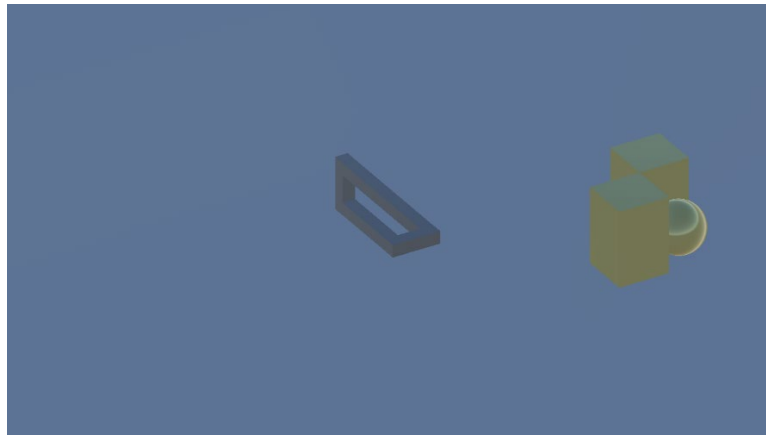


(c-2) 評価用動画c：提案手法による錯視立体と影， $\lambda = -\frac{\pi}{6}$

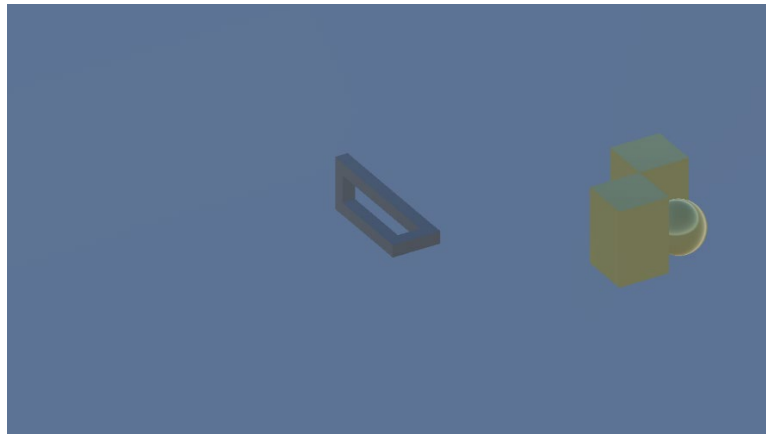
図28. 付影手法の主観評価実験に用いた動画



(a-3) 評価用動画a：影なし， $\lambda = -\frac{\pi}{3}$

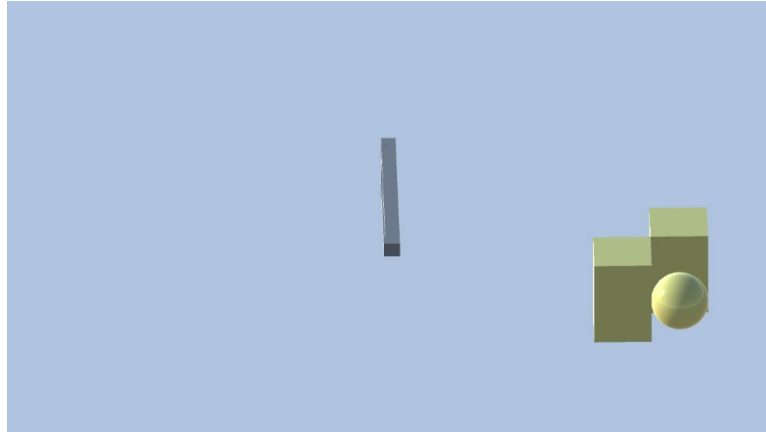


(b-3) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = -\frac{\pi}{3}$

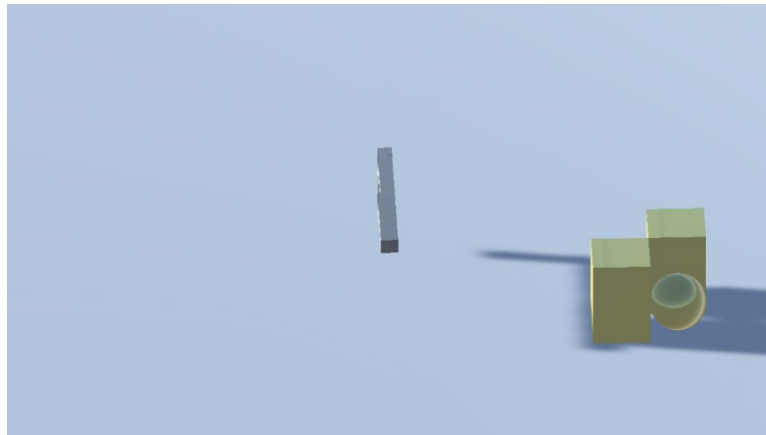


(c-3) 評価用動画c：提案手法による錯視立体と影， $\lambda = -\frac{\pi}{3}$

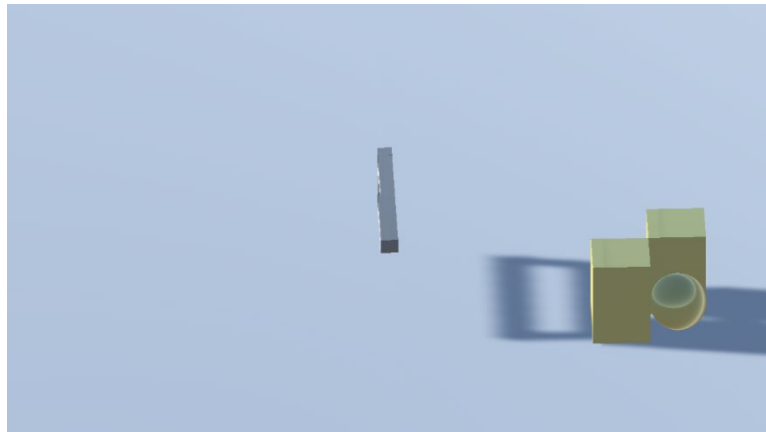
図28. 付影手法の主観評価実験に用いた動画



(a-4) 評価用動画a：影なし， $\lambda = -\frac{\pi}{2}$

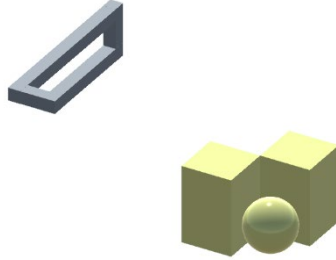


(b-4) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = -\frac{\pi}{2}$

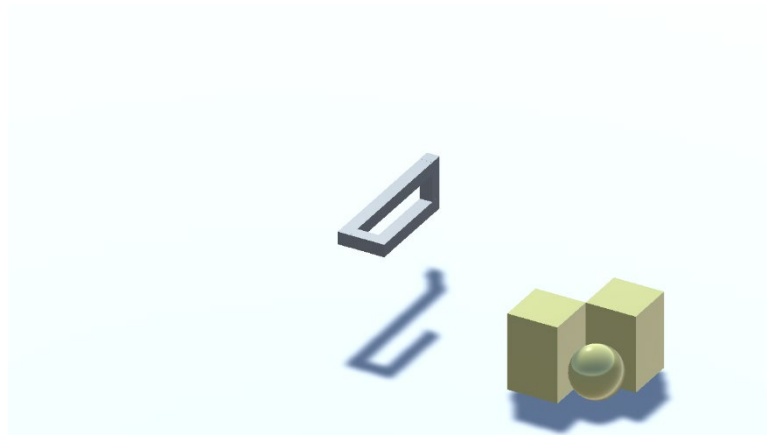


(c-4) 評価用動画c：提案手法による錯視立体と影， $\lambda = -\frac{\pi}{2}$

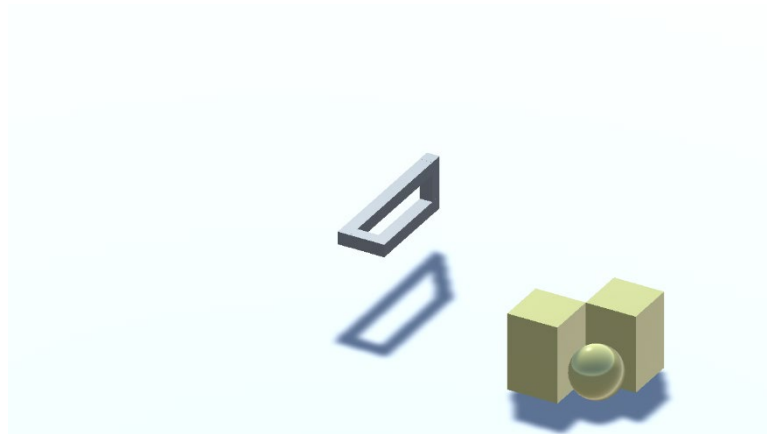
図28. 付影手法の主観評価実験に用いた動画



(a-5) 評価用動画a：影なし， $\lambda = -\frac{2\pi}{3}$

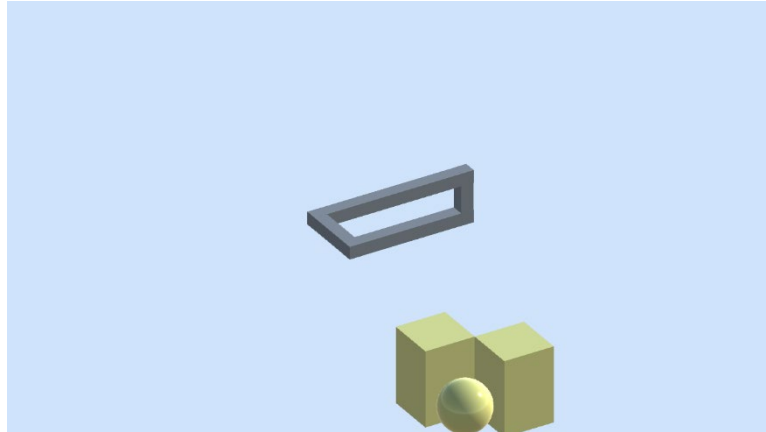


(b-5) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = -\frac{2\pi}{3}$

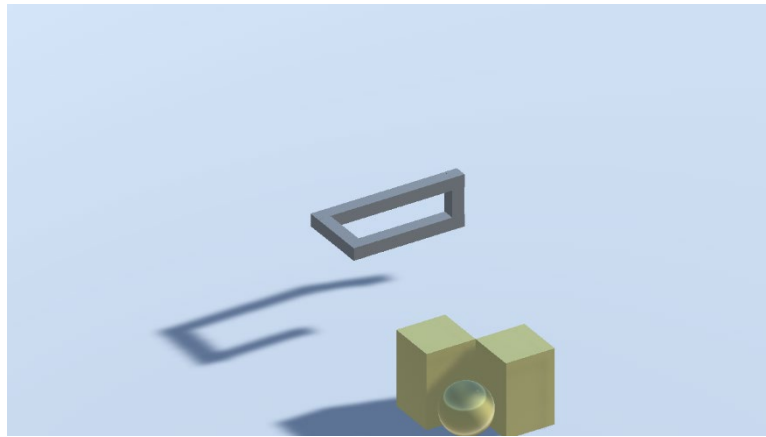


(c-5) 評価用動画c：提案手法による錯視立体と影， $\lambda = -\frac{2\pi}{3}$

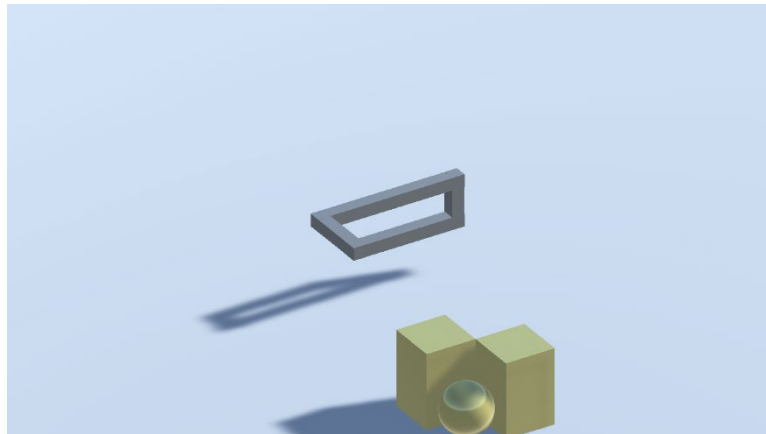
図28. 付影手法の主観評価実験に用いた動画



(a-6) 評価用動画a：影なし， $\lambda = -\frac{5\pi}{6}$

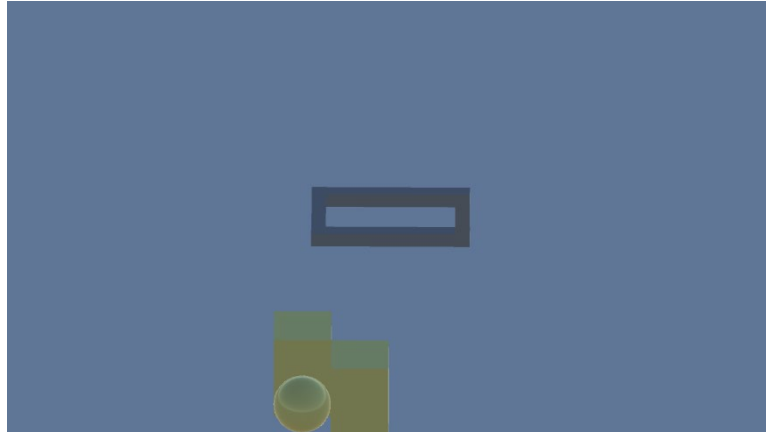


(b-6) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = -\frac{5\pi}{6}$

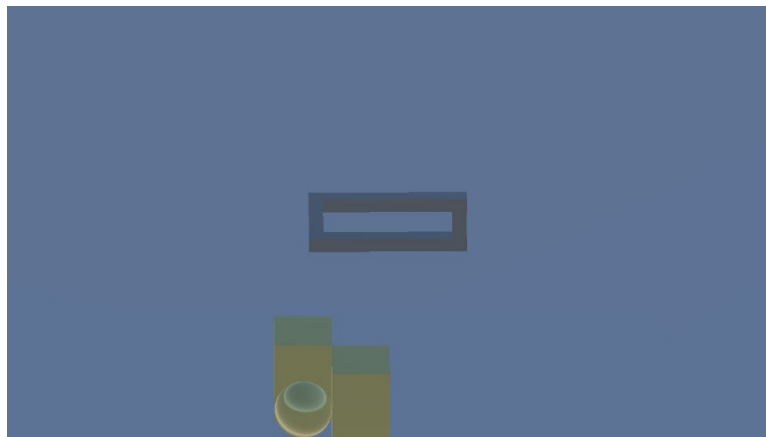


(c-6) 評価用動画c：提案手法による錯視立体と影， $\lambda = -\frac{5\pi}{6}$

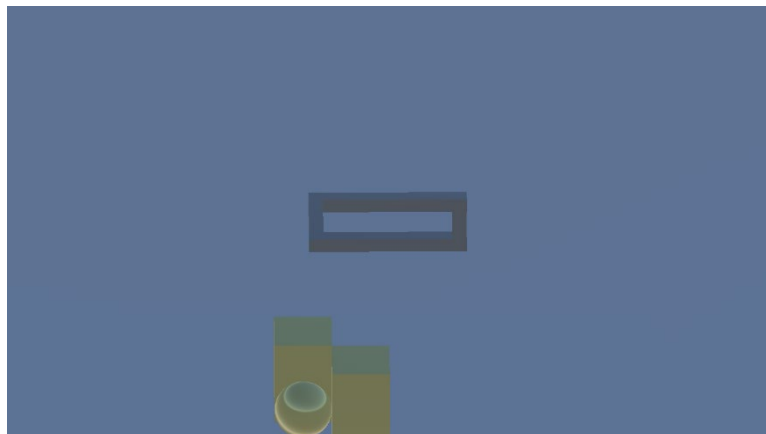
図28. 付影手法の主観評価実験に用いた動画



(a-7) 評価用動画a：影なし， $\lambda = -\pi$

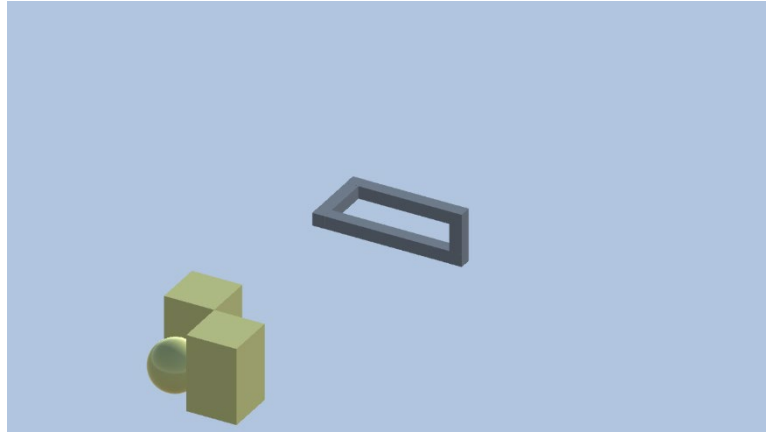


(b-7) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = -\pi$

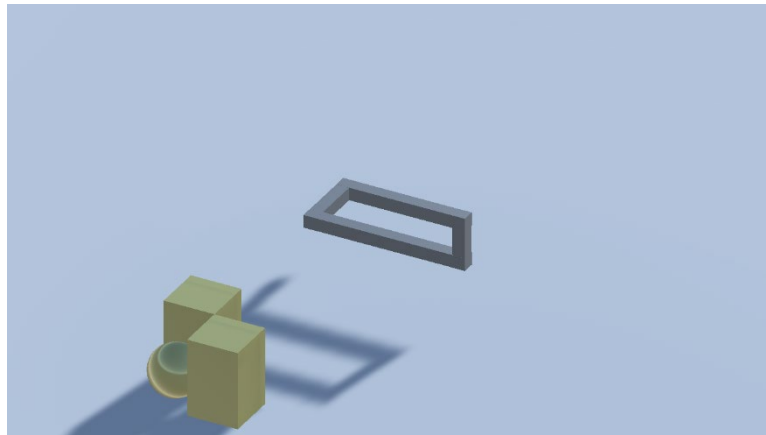


(c-7) 評価用動画c：提案手法による錯視立体と影， $\lambda = -\pi$

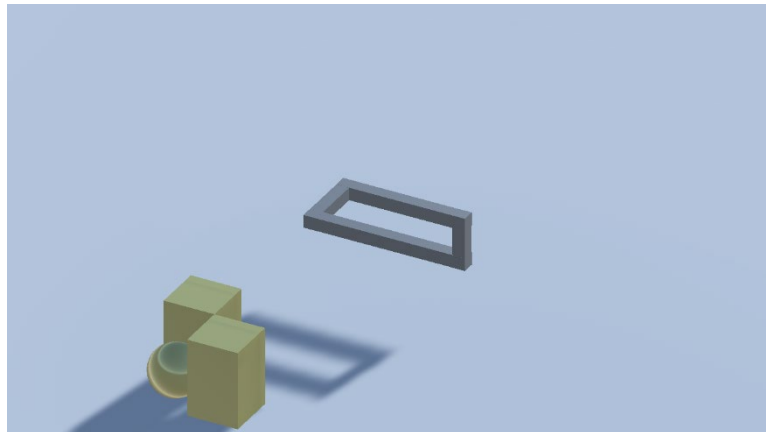
図28. 付影手法の主観評価実験に用いた動画



(a-8) 評価用動画a：影なし， $\lambda = \frac{5\pi}{6}$

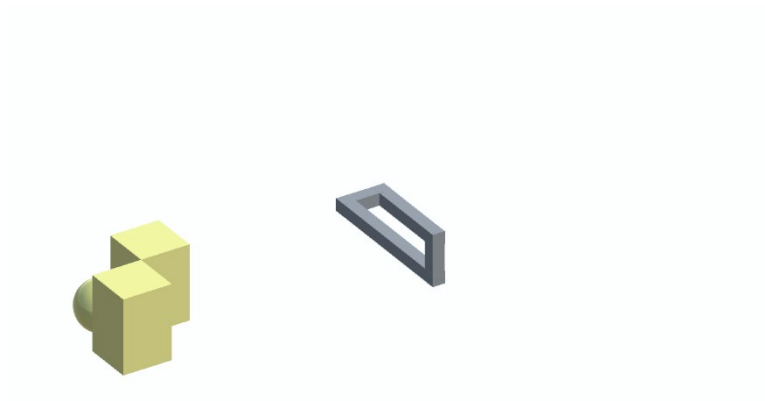


(b-8) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = \frac{5\pi}{6}$

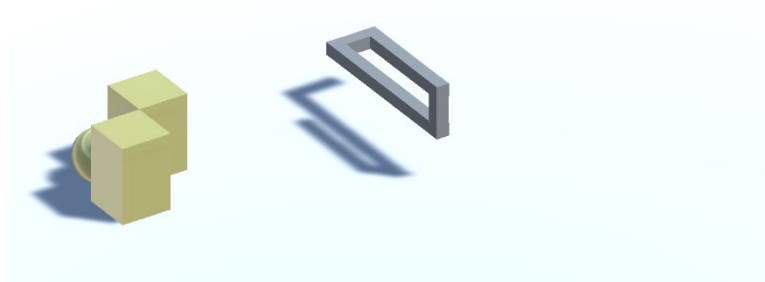


(c-8) 評価用動画c：提案手法による錯視立体と影， $\lambda = \frac{5\pi}{6}$

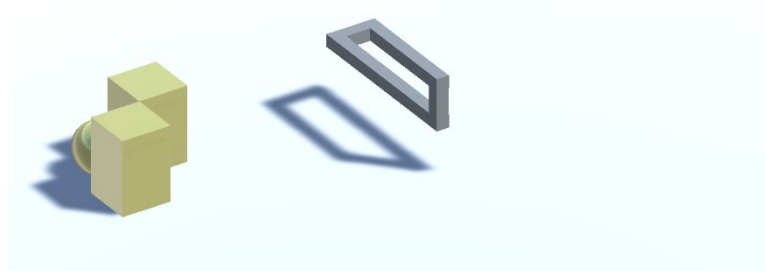
図28. 付影手法の主観評価実験に用いた動画



(a-9) 評価用動画a：影なし， $\lambda = \frac{2\pi}{3}$

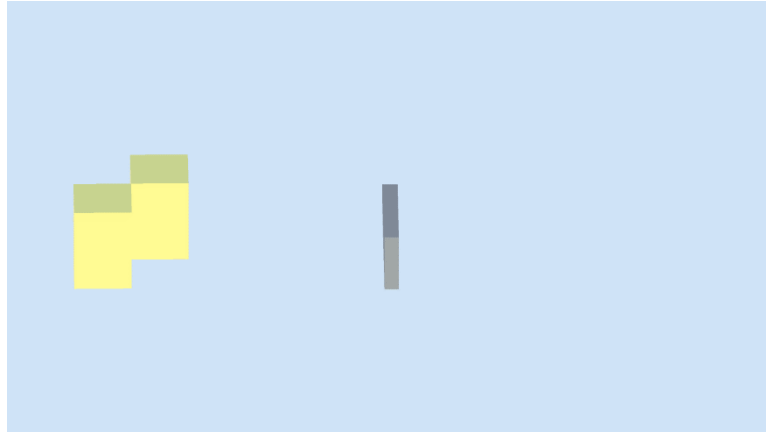


(b-9) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = \frac{2\pi}{3}$

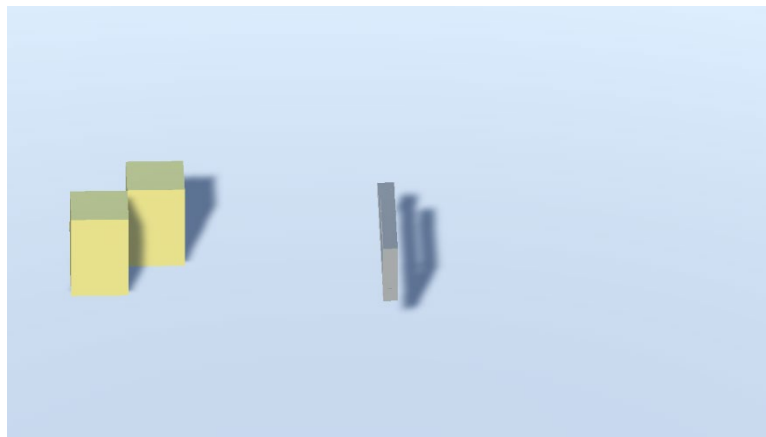


(c-9) 評価用動画c：提案手法による錯視立体と影， $\lambda = \frac{2\pi}{3}$

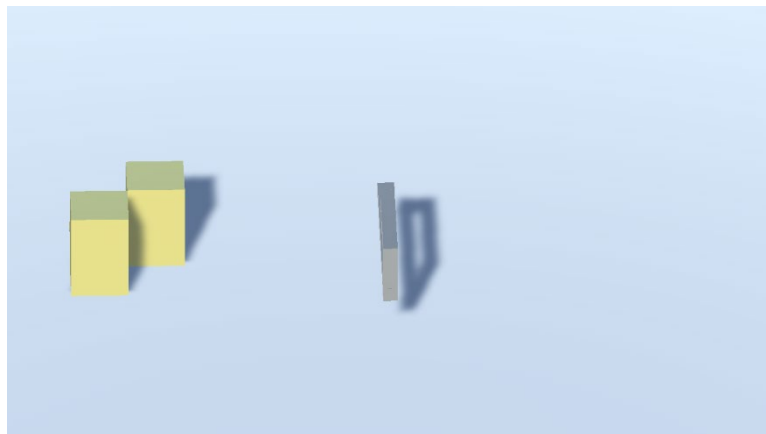
図28. 付影手法の主観評価実験に用いた動画



(a-10) 評価用動画a：影なし， $\lambda = \frac{\pi}{2}$

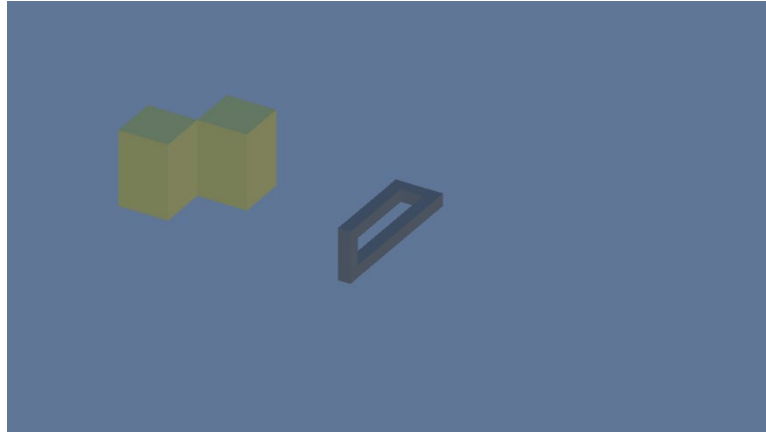


(b-10) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = \frac{\pi}{2}$

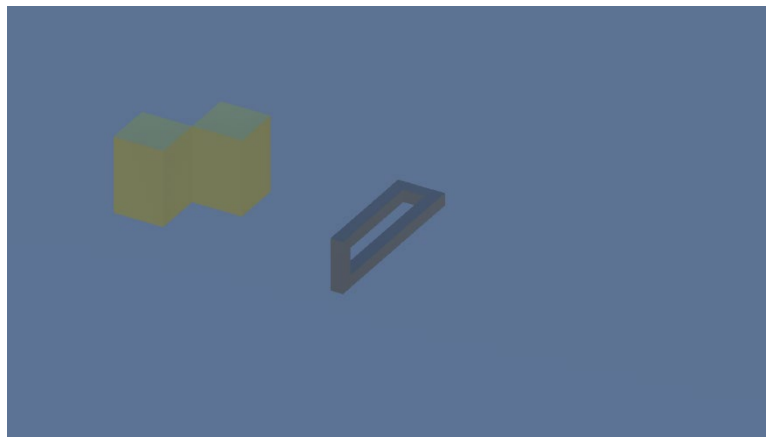


(c-10) 評価用動画c：提案手法による錯視立体と影， $\lambda = \frac{\pi}{2}$

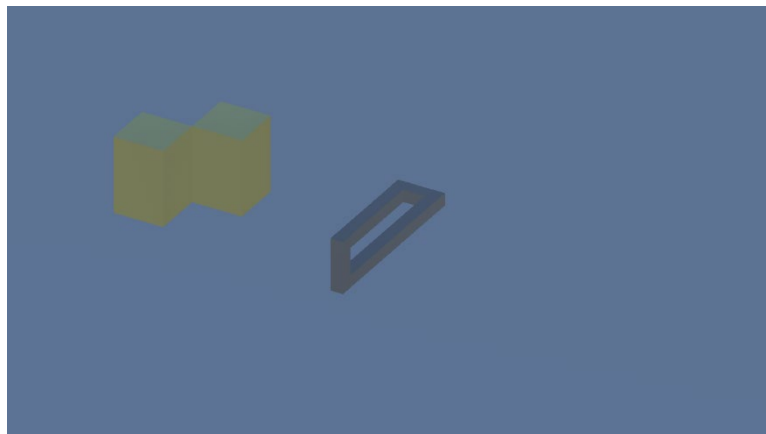
図28. 付影手法の主観評価実験に用いた動画



(a-11) 評価用動画a：影なし， $\lambda = \frac{\pi}{3}$

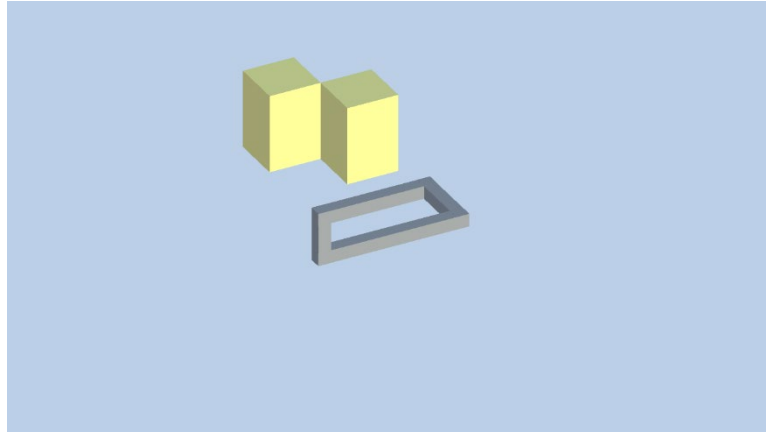


(b-11) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = \frac{\pi}{3}$

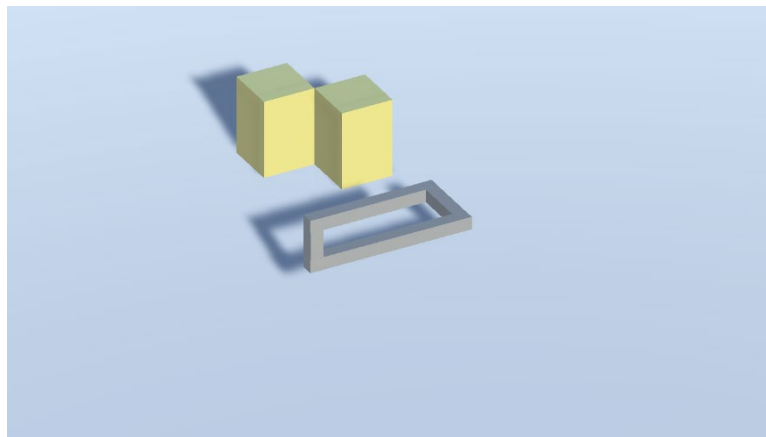


(c-11) 評価用動画c：提案手法による錯視立体と影， $\lambda = \frac{\pi}{3}$

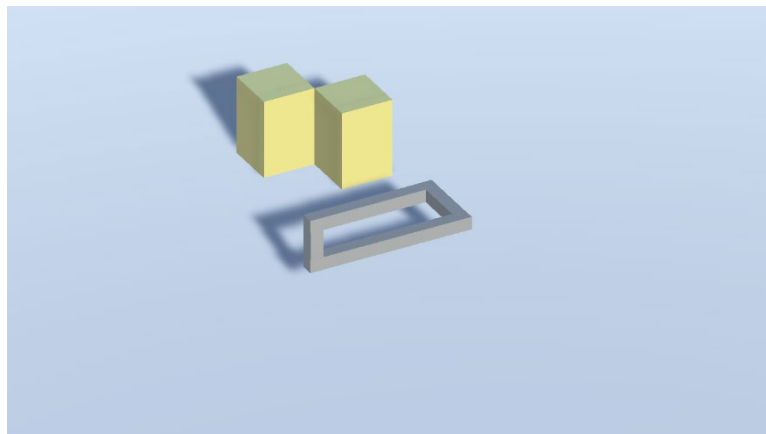
図28. 付影手法の主観評価実験に用いた動画



(a-12) 評価用動画a：影なし， $\lambda = \frac{\pi}{6}$



(b-12) 評価用動画b：不可能図形の印象とは異なる影， $\lambda = \frac{\pi}{6}$



(c-12) 評価用動画c：提案手法による錯視立体と影， $\lambda = \frac{\pi}{6}$

図28. 付影手法の主観評価実験に用いた動画

5.4.3 評価項目・評価方法

動画aを視聴した後、以下の問1、問2に、動画b, cは問1～問5について、「全くそう思わない」「どちらかと言うとそう思う」「どちらでもない」「どちらかと言うとそう思わない」「とてもそう思う」の5段階で評価させた。影なしの動画aは、影に関する問3と、動画aと比較して回答する問4と問5は設問から外した。

- | |
|---|
| 問1. 物体は不可能図形として成立しているように見える。 |
| 問2. この物体 (錯視立体)と横の黄色の物体群は同じ空間に存在するように見える。 |
| 問3. 下の影はこの物体の影だと思える。 |
| 問4. 影なしの動画aと比較して、不可能図形としての不思議さは強まった。 |
| 問5. 影なしの動画aと比較して、物体の存在感は強まった。 |

回答時間に制限は設けなかった。質問内容は動画を見た後に提示されるが、設問後に動画を再生しなおすことも可能とした。

5.5 付影処理の考察

上記主観評価実験結果を表2に示す。

表2 付影手法の主観評価実験結果

問	評価	評価				
		とてもそう思う	どちらかと言う とそう思う	どちらでもない	どちらかと言う とそう思わない	全くそう思わ ない
問1	影なし	9	15	0	1	0
	元の形状の影	3	16	2	3	1
	提案手法の影	12	11	1	1	0
問2	影なし	4	10	4	3	4
	元の形状の影	8	13	2	1	1
	提案手法の影	15	9	0	1	0
問3	元の形状の影	0	4	2	10	9
	提案手法の影	15	8	0	2	0
問4	元の形状の影	5	9	3	6	2
	提案手法の影	6	9	7	3	0
問5	元の形状の影	5	12	5	3	0
	提案手法の影	13	11	1	0	0

5.5.1 影が他の3DCGモデルとの共存感に及ぼす影響

問1「物体は不可能図形として成立しているように見える」は、影の描画が錯視効果に変化を与えるかどうかを検証する設問である。影を描画することによって、より錯視の効果を強め、不可能立体の印象をイメージしやすくなる可能性がある。その一方で、図26や図28(b)のように、錯視立体の仕組みを推測できてしまうような影を描画した場合、錯視効果が減少し、不可能立体の印象を損なうという仮説も成り立ち得る。問1の結果から、図28(b)を提示した際には、影なしと比較して「とてもそう(不可能図形として成立しているように)思う」割合が9名(36%)から3名(12%)に大きく減少した。図16(b)のように、本体から想像される不可能立体からかけ離れた形状の影を描画すると、不可能立体の印象は著しく減少するので避けなければならない。

一方、提案したShadow Illusion Objectsを用いて影を描画した図28(c)では、「とてもそう思う」人が12名(48%)に増加した。このことから、前述の仮説のとおり、影の描画が錯視効果に影響を与えることがわかった。

問2「この物体(錯視立体)と横の黄色の物体群は同じ空間に存在するように見える」は、付影処理によって、錯視立体と他の3DCGモデルとが共存しているように感じられるかについ

での設問である。ここではそれを共存感と呼ぶことにする。実際に共存しているような印象を与えるか、影によって共存しているような印象がどう変化するかを検証した。

問2の結果は、動画a, b, cの順、つまり図28(a), 図28(b), 図28(c)の順に「とてもそう思う」割合が4名(16%), 8名(32%), 15名(62.5%)となった。すなわち、錯視立体と他の3DCGモデルが同時に映っている画面に影を加えると、共存感が増す。この共存感は図28(b)でも観察されたが、図28(c)に示すように、提案したShadow Illusion Objectsを用いることで、錯視立体と他の3DCGモデルが共存していることを強く印象づけることができた。

問1, 問2の結果により、提案したShadow Illusion Objectsを用いた付影処理は、錯視効果に大きな影響を与えること、錯視立体に他の3DCGモデルとの共存感を強く与えることがわかった。

5.5.2 錯視立体の影と不可能立体の関係の知覚

問3「下の影はこの物体の影だと思える」は、描画した影が要件③を満たすかどうか、すなわち、Shadow Illusion Objectsを用いて描画した影が、不可能立体を本体とする影として知覚されるかどうかを検証する設問である。

問3の結果は最も大きな差が出ており、図16(b)では「とてもそう思う」が0%であった一方、回答の76%が「全くそう思わない」または「どちらかと言うとそう思わない」であり、錯視立体そのものの影は、不可能立体を本体とする影として知覚されていなかった。逆に、図28(c)では回答の92%が「とてもそう思う」「どちらかと言うとそう思う」であり、Shadow Illusion Objectsを用いる提案手法は、不可能立体を本体とする影として知覚されることがわかった。

本論文で提案した拡張表現モデルを用いたことで、このモデルの特質である、印象のロバスト性が問3の結果に影響を与えたと推測される。光源から見たときの錯視立体のシルエットが、カメラ視点から見た不可能図形と同じ不可能立体であるという印象を与えたので、影の本体は錯視立体であろうと知覚されたと考えられる。

5.5.3 トリックアートの面白さへの影の寄与

問4「影なしの動画aと比較して、不可能図形としての不思議さは強まった」、問5「影なしの動画aと比較して、物体の存在感は強まった」は、不可能図形の面白さを向上させるのに影がどのように影響するかを検証する設問である。第1章で述べたように、不可能図形の面白さは、不可能立体をイメージできるという存在感と、実際にはそれが実在しないという不思議さの矛盾からなる。ここで、その不思議さと存在感、それぞれの尺度から質問を設けた。

問4の結果は、図28(b), 図28(c)で「とてもそう思う」「どちらかと言うとそう思う」の人数

にはほぼ変化がなかった。つまり、影さえ表示されていれば、それがどのような形をしたものであれ、トリックアートとしての違和感が増し、不思議さは強まるという結果であった。これは、不可能図形がもともと違和感による不思議さを楽しむものであるため、多少理屈に合わない形であっても、同じ違和感の中で扱う心理が働いたためと考えられる。

問5の結果は、図28(b)で20%、図28(c)で52%の被験者が「とてもそう思う」と回答した。問2と同じく、影が存在するだけで錯視立体の存在感は強まるが、提案手法を用いることでその効果をさらに高めることができた。

問4と問5の結果から、不可能図形の不思議さと存在感は、それぞれ独立した性質であり、提案したShadow Illusion Objectsを用いることで、その双方の相乗効果を期待できると考えられる。

5.5.4 考察のまとめ

提案したShadow Illusion Objectsを用いて影を描画することで、不可能図形が他の3DCGモデルと共存しているという印象や、不可能図形自体の存在感を一段と向上させることができた。そのため、提案手法は、3DCG作品中に不可能図形アニメーションを組み込む際に、非常に効果的であると期待される。

第6章

結論

6.1 本論文のまとめ

本研究では、不可能図形という種類のトリックアートをアニメーションさせ、3DCG作品中に組み込むことを目的とした。特定の視点から観察すると不可能図形として知覚される立体、すなわち錯視立体に着目し、他の一般的な3DCGモデルと共存可能な錯視立体表現モデル（拡張表現モデル）を提案した。

また、より自然に不可能図形アニメーションを3DCG作品中に組み込むために、錯視立体の付影処理手法を提案した。

本研究の特徴、および、本研究によって明らかになった点を以下にまとめる。

(1) 他の3DCGモデルと共存可能な錯視立体表現モデル(拡張表現モデル)

錯視立体の3DCGモデルと、視点の移動に合わせて錯視立体を動的に再モデリングするアルゴリズムからなる、錯視立体表現モデルを提案した。

提案した錯視立体表現モデルは、汎用レンダリング環境で実装可能である。すなわち、研究者ではない3DCGアーティストなどが、他の一般的な3DCGモデルに向けて作成した作品中の要素として、不可能図形のアニメーションを利用することが可能である。特に、汎用レンダリング環境の例として、Unity^[11]を使用した実装を示した。Unityを用いることで、ゲーム、映像作品のみならず、昨今進歩著しいAR/VR等のインタラクティブコンテンツの多くで不可能図形アニメーションを表示可能となった。

また、提案した錯視立体表現モデルは、シンプルかつ自動化された錯視の仕組みを持ち、変形しても印象のロバスト性が高いという特徴をもつ。これにより、他の3DCGモデルと不可能図形アニメーションを共存させたコンテンツの制作が可能となった。

(2) 不可能立体の影の在り方に基づく付影処理手法

不可能立体と他の3DCGモデルの共存感を向上するために、錯視立体の付影処理手法を提案した。

印象上にしか存在しない不可能立体の影の在り方についての考察を踏まえ、錯視立体の影が満たすべき要件を明らかにした。要件に基づいて、錯視立体の影を定義し、Shadow Illusion Objects法を提案した。

また、錯視立体の影に関する主観評価実験を行った。錯視立体に影を付与することで、不可能立体が他の3DCGモデルと共存しているという印象や、不可能立体自体の存在感を一段と向上させることができた。さらに、提案手法の影は、表示されている錯視立体から落ちた影であると正しく知覚される。そのため、提案手法は、3DCG作品中に不可能図形アニメーションを組み込む際に、非常に効果的であると期待される。

6.2 今後の展開

本節では、本研究の発展の可能性と、それにより生じるであろう課題について述べる。

6.2.1 視点の特異点の除去

提案手法において、例えば図29のように視点が極座標の極付近にあるとき、平行な角柱#2と角柱#4を、角柱#1で接続しているように見せかけることと、角柱#3と角柱#1が垂直となることの2点を満たすように変形するため、角柱#1の長さが無限大に発散してしまい、不可能図形の形状として破綻してしまう。図29の例では角柱#1がカメラのクリッピング距離を超えて視点の後ろ側まで伸びているため、端が描画されなくなっている。加えて、角柱#3と角柱#1の位置関係も破綻している。このような視点の特異点を除去可能な変形方法について検討をすすめる。

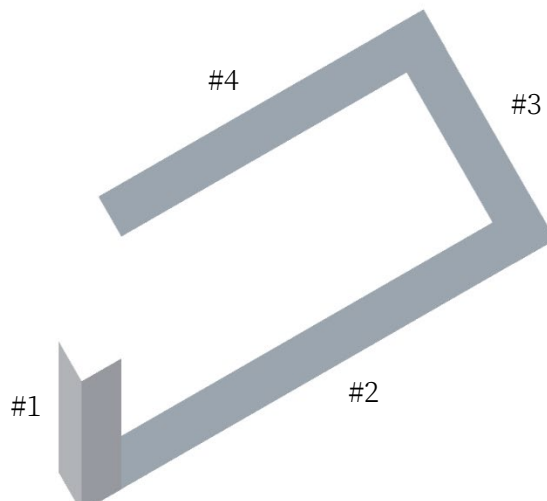


図29. 破綻した錯視立体の例

6.2.2 別の種類の不可能図形での表現

4.4節において、トーラス状不可能図形と同じ錯視の仕組みである、無限にループする水路など、より複雑な形状への適用例を示した。中津ら^[4]によれば、トーラス状不可能図形の錯視立体表現モデルは、図30に示したペンローズの階段^[18]のようなモデルに対しても、適用可能である。しかし、中津ら^[4]の手法によるペンローズの階段のアニメーションは、錯視立体の部分立体どうしの稜線の接続を応用することで実現されている。これは、第4章で述べたとおり、サーフェスモデルで実装することができないため、他の3DCGモデルと共存させることができない。

提案した錯視立体表現モデルのように、一般的な3DCGモデルと共存可能な錯視立体表現モデルを、さらに多彩な不可能図形アニメーションに対応させるため、検討を進める。

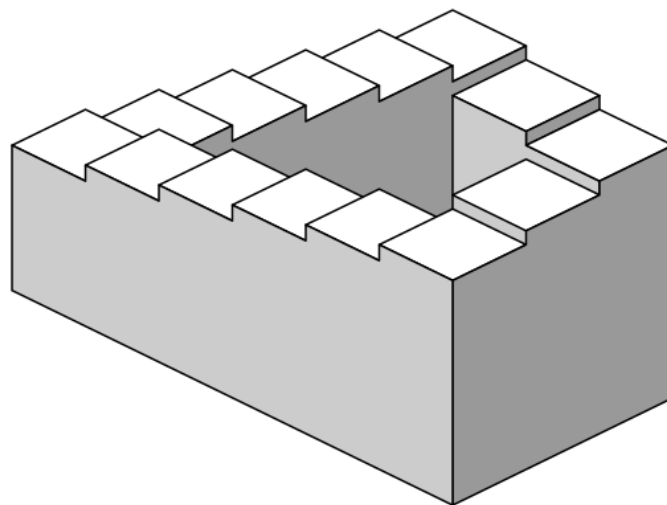


図30. ペンローズの階段^[18]

6.2.3 付影処理の高度化

本論文で提案した付影処理手法は，錯視立体から他の物体へ影を落とすことのみを扱っている．逆に，他の物体から落ちる影を錯視立体に落とすことができれば，観察者が立体構造を把握する手がかりになる．

第5章で述べたとおり，立体構造把握の手がかりは，錯視効果の向上に寄与する可能性があるが，図31のように，錯視立体の仕組みを露呈してしまい，錯視効果を減少させてしまう可能性もある．そのため，Self-Shadow を含む，付影処理の高度化については，慎重に検討を重ねる必要があり，今後の課題である．

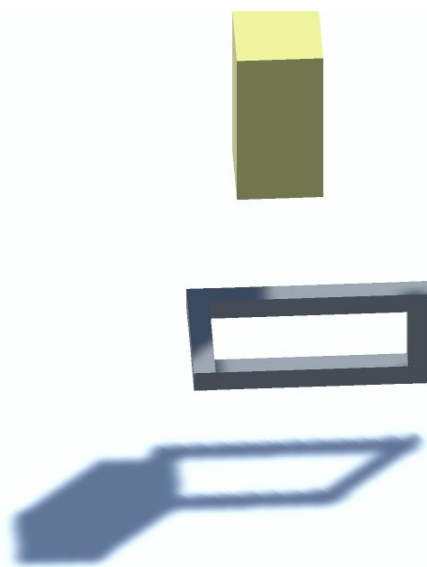


図31. 錯視立体の仕組みを露呈してしまう影付けの例

謝辞

本研究を進めるにあたり，指導教員として，熱心にご指導，ご鞭撻いただきました東京電機大学 未来科学部 情報メディア学科 高橋 時市郎 教授に深く感謝いたします。

東京電機大学 未来科学部 情報メディア学科 森谷 友昭 准教授には，副査および副指導教員として多くのアドバイスを頂きました。記して感謝いたします。

東京電機大学 未来科学部 情報メディア学科 鉄谷 信二 教授，ならびに，工学部 情報通信工学科 長谷川 誠 教授には，本論文の副査としてご助言を頂きました。記して感謝いたします。

東京電機大学 ビジュアルコンピューティング研究室の中津 香奈さん（現 NTTデータ）には，多くのアドバイスや先行研究の詳細なアルゴリズムの提供を頂きました。記して感謝いたします。

また，本研究に対して，熱心にご意見頂きました，東京電機大学 ビジュアルコンピューティング研究室の皆様，特に，ミーティングを通して，活発に議論を交わしていただいた窪田 裕介さん（現 富士通Japanソリューションズ東京），森 政晴さん（現 NTTテクノクロス），山邊 悠太さん（現 東京医療保健大学 助教），吉田 大さん（現 日立製作所），小玉 周平さん（現 東京電機大学 助教）に，深く感謝いたします。

最後に，博士課程在学中に住居面でサポートしてくださった，祖父である，東京工業大学 千葉 廉 名誉教授に感謝いたします。

参考文献

- [1] 杉原 厚吉: “不可能物体の数理,” 森北出版株式会社 (1993).
- [2] M. C. Escher: “Waterfall,” 300mm × 380mm, Lithograph (1961).
- [3] 中津 香奈, 森谷 友昭, 高橋 時市郎: “錯視立体を用いたトラス状不可能図形のアニメーション手法,” 画像電子学会誌, Vol.45, No.3, pp.359-369 (2016).
- [4] T. P. Wu, C. W. Fu, S. K. Yeung, J. Jia, C. K. Tang: “Modeling and Rendering of Impossible Figures,” ACM Trans. On Graphics, Vol.29, No.2, pp.13:1-13:15 (2010).
- [5] C. W. Khoh, P. Kovesi: “Rotating the Impossible Rectangle,” Leonardo, Vol. 34, No. 3, pp. 197-198 (2001).
- [6] S. Owada, J. Fijiki: “DynaFusion: A Modeling System for Interactive Impossible Objects,” NPAR '08, pp. 65-68 (2008).
- [7] 藤木 淳, 大和田 茂: “theRelativity 一人称視点と三人称視点の見え方の違いを互いに反映するだまし絵表現技法,” 第4回デジタルコンテンツシンポジウム (2008).
- [8] J. Fujiki, S. Owada: “theRelativity”, ACM SIGGRAPH ASIA 2008 Art Gallery (2008).
- [9] 篠原 祐樹, 宮下 芳明, “不可能立体の写実的表現手法の提案,” HCI, Vol. 2009, No. 28, pp. 95-102 (2009).
- [10] 吉川 祐輔, 宮下 芳明: “画像バッファの組み合わせによるリアルタイム錯覚表現生成手法,” 芸術科学会論文誌, Vol.12, No.1, pp.11-23 (2013).
- [11] Unity Technologies Inc., Unity, <https://unity.com/> (2021).
- [12] Impossible trident, Wikipedia (en), https://en.wikipedia.org/wiki/Impossible_trident (2021).
- [13] File:Blivet2.svg, Wikipedia Commons, Public Domain, <https://commons.wikimedia.org/wiki/File:Blivet2.svg>

imedia.org/wiki/File:Blivet2.svg (2021).

- [14] Kinect for Windows, Microsoft, <https://developer.microsoft.com/ja-jp/windows/kinect/> (2021).
- [15] J. Birn, Digital Lighting and Rendering, New Riders, pp.106-107 (2013).
- [16] コンピュータグラフィックス編集委員会: “コンピュータグラフィックス [改訂新版],” 公益財団法人画像教育振興協会 (CG-ARTS協会), pp.158-161 (2015).
- [17] シャドウマッピング, Unity Manual, <https://docs.unity3d.com/ja/2021.2/Manual/shadow-mapping.html> (2021).
- [18] File:Impossible staircase.svg, Wikipedia Commons, Public Domain, https://commons.wikimedia.org/wiki/File:Impossible_staircase.svg (2021).