

## 逆遠近感を利用した錯視立体図形のモデリング

伊藤 紘治<sup>1)</sup> 金森 由博<sup>1)</sup>(正会員)  
三谷純<sup>1) 2)</sup>(正会員) 福井 幸男<sup>1)</sup>(正会員)

1) 筑波大学 2) JST ERATO

## Modeling of Reverse Perspective Objects

Hiroharu Ito<sup>1)</sup> Yoshihiro Kanamori<sup>1)</sup>  
Jun Mitani<sup>1) 2)</sup> Yukio Fukui<sup>1)</sup>

1) University of Tsukuba 2) JST ERATO

itoh@npal.cs.tsukuba.ac.jp, {kanamori,mitani,fukui}@cs.tsukuba.ac.jp

### 概要

逆遠近感を利用した錯視立体図形は、観察者が見る位置を変えることによって、物体が静止しているにもかかわらず、変形しているかのように見える立体である。本研究では、三次元ポリゴンモデルを入力として、逆遠近錯視立体をモデリングする手法を提案する。ユーザが適当な視点の位置を決めると、システムはその視点から可視である面について、奥行きが反転するようにモデルを適切に変形させ、逆遠近錯視のモデルを生成する。また、ユーザがモデルの一部を指定することで、指定した部分だけを逆遠近錯視立体へ変形させることも可能とする。さらに本研究では、得られたモデルをペーパークラフトとして実世界に出力し、実際に錯視が起きることを確認した。

### Abstract

A reverse perspective object is a 3D shape that seems as if it were deformed when the viewer moves. We propose a method for modeling a reverse perspective object from a 3D polygonal model as an input. Given a user-specified viewpoint, the system deforms visible faces of the input model appropriately so that the depths of the faces are flipped, and then outputs a reverse perspective object. In addition, our method can deform not only the whole shape of the model but also user-selected parts of the shape to generate a reverse perspective object. Furthermore, we also confirm that the output models actually play tricks on human eyes by making them as paper crafts.

## 1 はじめに

逆遠近感を利用した錯視（逆遠近錯視）は、観察者が見る位置を変えることによって、観察している物体が静止しているにもかかわらず、変形しているかのように見える錯視である。逆遠近錯視はトリックアートの一つとして広く親しまれており、既存の作品例として図1に示すドラゴンのペーパークラフト [4] や図2に示す風景画の立体 [6] などがよく知られている。逆遠近錯視は、物体の実際の凹凸と脳が解釈している凹凸が逆のときに起こる。そのような凹凸を持つ物体を注視しながら移動すると、脳が期待する見え方の変化とは異なる変化が観察され、脳がその矛盾を解消しようとした結果、「物体が変形している」という錯覚が起きる。

このような逆遠近錯視を引き起こす立体を簡易にモデリングできれば、逆遠近錯視をより身近なトリックアートとして鑑賞でき、また逆遠近錯視の研究にも貢献できると思われる。しかし、既存のソフトウェアでは逆遠近錯視立体のモデリングは容易でない。そこで本研究では、三次元のポリゴンモデルを入力として、逆遠近錯視立体をモデリングする手法を提案する。ユーザが適切な視点の位置を決めると、システムはその視点からのモデルの見え方をできるだけ保ったまま、奥行きが反転するようにモデルを適切に変形させ、逆遠近錯視のモデルを生成する。また、図1のドラゴンは頭部が逆遠近錯視立体で胴体が通常の立体となっており、本研究ではこのような作品も作成できるようにする。すなわち、ユーザがモデルの一部分を指定することで、指定した部分だけを逆遠近錯視立体へ変形させることも可能とする。提案手法を実装したシステムを用いることで、図3のような結果を簡単に作成できる。さらに本研究では、得られたモデルをペーパークラフトとして実世界に出力し、実際に錯視が起きるのか確認を行った。

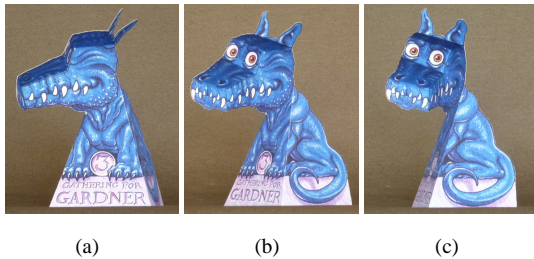


図1: 逆遠近錯視の作品例（ドラゴン）: (a) 正面から撮影, (b)(c) カメラを移動させて撮影。視点の位置に合わせて頭部がこちらを向くように回転しているように見える。実は頭部だけ凹んでいて錯視が起きるようになっている。Copyright ©1988 BINARY ARTS CORPORATION.

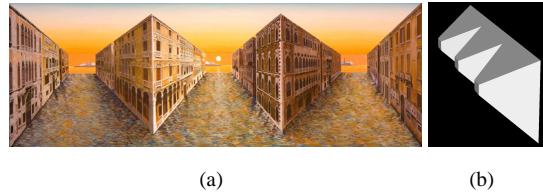


図2: 逆遠近錯視の作品例（風景画）: (a) 正面から見たときは単なる絵画に見える。観察者が左右に移動すると建物が動いているように見える。(b) 斜め上方から見ると錐台が三つ連なったような形状をしていることがわかる。建物ではなく背景の部分の部分が出っ張っているため、錯覚が起こる。Copyright ©2010 Patrick Hughes.

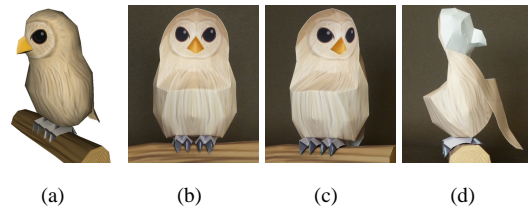


図3: 提案手法の適用例（フクロウ）: (a) の入力モデルの頭部だけを変形させると、(b)(c)(d) のような立体を生成できる。(b)(c) のように視点を変えて観察すると、視点の位置に合わせて頭部が回転しているように見える。(d) のように横から見ると、実際の凹凸がわかる。

## 2 関連研究

錯視を研究することは、人間の視覚機能を調べることでもある。そのため、錯視の研究はコンピュータグラフィックス (CG) やコンピュータビジョン (CV) の分野だけでなく、心理学や脳科学の分野でも行われている。

### 2.1 心理学・脳科学における錯視の研究

心理学・脳科学の分野では、さまざまな種類の錯視が研究されており、逆遠近錯視もしばしば研究の対象になっている。逆遠近錯視の研究は、心理学の分野では Cook ら [1][2][3] や Papathomas [9] によって、脳科学の分野では Hayashi ら [5] によって行われており、逆遠近錯視はホロウマスク錯視と同じ原理で起こることが知られている。

ホロウマスク錯視は、図4に示すように、凸面の仮面を裏側から見ても、凹面ではなく凸面に見えてしまうという錯視である。これは「仮面は凸面である」という経験則から、脳が自動的に凹凸を解釈するために起こる。そして、凹凸を誤って解釈した結果、逆遠近錯視と同様に、観察者の移動に合わせて仮面が回転しているかのように見える。



図 4: ホロウマスク錯視。仮面を裏側から見ると、本当は凹んでいるはずなのに、出っ張っているように見える。 Copyright ©1999 NTT COMMUNICATION SCIENCE LABORATORIES.

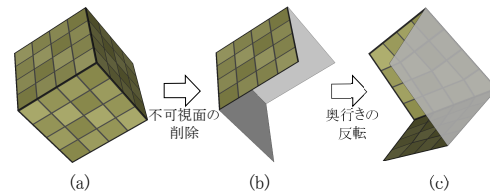


図 6: 基本手法の流れ。入力モデル (a) について基準視点から見えない面を削除し (b)、凹凸を反転させて逆遠近錯視立体を生成する (c)。基準視点は向かって左側にある。

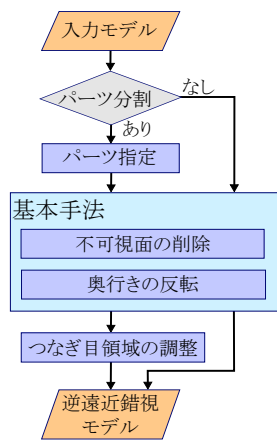


図 5: 提案手法の流れ。

## 2.2 CG・CV における錯視の研究

CG・CV 分野における錯視の研究は、錯視立体図形モデリング手法として、Wuらの手法 [13] や、OwadaとFujikiの手法 [8] などがある。また、Sugiharaの手法 [10] で錯視立体図形を生成できる。ただし、彼らの手法は不可能図形と呼ばれる種類の錯視立体図形を作るためのものであり、逆遠近錯視の立体は対象としていない。

逆遠近錯視の仕組みを数理的に解説した文献として、杉原の著書 [11] がある。逆遠近錯視のモデリングに関しては、画像を入力とする手法が高野 [12] によって提案されている。しかし、高野の手法で生成できるのは錐台形状の逆遠近錯視立体のみである。つまり図 2 と同種の逆遠近錯視立体のみ生成できる。本研究の提案手法は、入力がポリゴンモデルであり、高野の手法よりも多種多様な逆遠近錯視立体を生成できる。

## 3 逆遠近錯視立体のモデリング

本研究では、入力として三次元ポリゴンモデルを用い、これに変形を加えて逆遠近錯視立体をモデリングする手法を提案する。具体的には、入力モデルの各頂点の三次元座標を書き換えることにより、逆遠近錯視の立体を生成する。結果をペーパークラフトとして組み立てることを想定し、入出力はポリゴン数の少ないモデルとする。また、すべての面は三角形であるとする。

本節では、まず 3.1 節で、逆遠近錯視の立体を生成する基本手法について述べ、次に 3.2 節で、基本手法を発展させた手法として、パーツ分割を使った逆遠近錯視立体の生成手法について述べる。図 5 に、提案手法の全体の流れを示す。

### 3.1 凹凸を反転させる基本手法

逆遠近錯視の立体への変形は、基本的には、ホロウマスク錯視の仮面のように、ある視点から見た立体の凹凸を反転させればよい。なお、出力される立体の形状は、入力モデルと視点情報によって決まる。視点情報とはモデルを観察する視点の位置や画角のことで、これらはユーザにより決定される。以後、ユーザによって決められる視点のことを「基準視点」と呼ぶ。

提案手法では投影変換に基づいてモデルを変形させ、逆遠近錯視の立体を生成する。ただし、上記のような変形を行う際には次の二つのことに注意しなければならない。

第一に、基準視点から見えない位置にある（入力モデルの）面を削除した上で、変形を行う必要がある。見えない面を削除する理由は、それらの面が存在していると逆遠近錯視の立体へ変形させたとき、奥行きが反転することによって、本来見えてほしい面がそれら不要な面に遮られてしまうからである。

第二に、基準視点から見たとき、入力モデルと出力モデルができるだけ同じ形状に見えるようにする必要がある。これを実現するには、まず、三次元ポリゴンモデルをスクリーン上に投影し、入力モデルの各頂点に対応するスクリーン上での  $x, y$  座標と奥行きを調べる。そして、奥行きだけを書き換えてモデルの凹凸を

反転させることで、頂点のスクリーン上での  $x, y$  座標を保ったまま目的の形状が得られる。

本節ではまず、基準視点から入力モデルを見たときの不可視面を特定し削除する方法について述べる。次に、本節の要である、モデルの奥行きを反転させて逆遠近錯視の立体を生成する方法について述べる。基本手法の流れを図 6 に示す。

### 3.1.1 不可視面の削除

入力の三次元ポリゴンモデルを逆遠近錯視の立体に変形させる際には、基準視点から可視である面のみを対象とする必要がある。そのため、変形の前処理として、基準視点から不可視の面を削除する。

不可視面の特定には、デプステストによる隠面消去と、カラーバッファに擬似カラーを描画する手法 [7] を用いる。この方法では、入力モデルの各面に固有番号を割り振り、その番号を 24 ビット整数値として RGB 各 8 ビットのチャンネルにエンコードして、擬似カラーとして保持する。そしてデプステストを有効にし、擬似カラーを持つ面をカラーバッファに描画することで、可視面の擬似カラーだけを取得できる。これによって不可視面を特定し削除できる。

提案手法を実装したシステムでは、上記の不可視面を削除する方法の他に、ユーザが任意に削除する面と削除しない面を選択できるようにしている。この理由は、不可視面の自動削除時の不具合への対策として、実物の制作時の都合による。例えば、凹凸を反転させた際、観察したい部分を遮ることになる面は削除しておきたいが、その面が少しでも基準視点から見えていると削除されずに残ってしまう。また、生成したモデルをペーパークラフトなどの形で実世界に出力したい場合、基準視点から不可視の面を本来は削除すべきだが、紙で形状を維持できる構造にするために残しておきたいこともある。

### 3.1.2 凹凸の反転

奥行きを反転させて逆遠近錯視の立体に変形させる工程は、三つの段階に分けられる。図 7 に、各段階での座標変換や変形の流れを示す。第一段階として、前工程で削除しなかった入力モデルの面を対象として、面を構成している各頂点に対して座標変換を行いウィンドウ座標系に変換する。第二段階としてウィンドウ座標系で、面の凹凸が反転するように頂点の奥行き値を書き換える。第三段階として、反転後の頂点をワールド座標系へ戻し、逆遠近錯視の立体を生成する。

第一段階として、ワールド座標系の点  $\mathbf{p} = (x, y, z, w)^T$  に対応するウィンドウ座標系の点  $\mathbf{p}_w$  を考える。モデルビュー行列を  $\mathbf{M}$ 、透視投影行列を  $\mathbf{P}$ 、ビューポート行列を  $\mathbf{V}$  とすると  $\mathbf{p}_w$  は次のように書ける。

$$\mathbf{p}_w = \mathbf{VPM}\mathbf{p} \quad (1)$$

第二段階はウィンドウ座標系で行う。まず、各点の奥行き値を参照して反転の基準となる奥行き値を求める。そしてその基準となる奥行きに対し、各点の奥

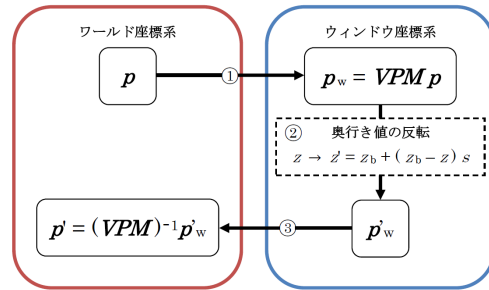


図 7: 逆遠近錯視立体を生成するための座標変換の流れ。①ワールド座標系からウィンドウ座標系へ座標変換、②ウィンドウ座標系で奥行き値を反転、③ウィンドウ座標系からワールド座標系へ逆座標変換。

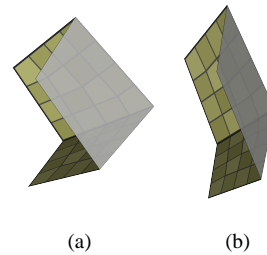


図 8: 奥行きをの圧縮。(a) スケーリング係数が 1, (b) スケーリング係数が 0.5。モデルは図 6 のものと同じで、基準視点は向かって左側にある。

行きを反転させる。元々の奥行き値を  $z$  とすると、新しい奥行き値  $z'$  は次の式ようになる。

$$z' = z_b + (z_b - z)s \quad (2)$$

ここで  $z_b$  は反転基準の奥行き値、 $s \in [0, 1]$  は奥行き値のスケーリング係数である。

反転の基準となる奥行き  $z_b$  は、すべての頂点の奥行き値のうちの中央値を用いる。例えば反転基準として平均値を用いると、極端な奥行きの変化がある場合に反転基準の奥行きがモデルの中心から離れてしまい、ユーザの意図しない変形が起こりやすくなる。中央値であればそのような極端な奥行きの影響を減らすことができる。スケーリング係数  $s$  は、奥行きをどれだけ圧縮するか決めるもので、通常は  $s = 1$  とする。スケーリング係数を調整すると、背面が見えずに鑑賞できる角度が広がる。例えば図 8(a) では、観察者が左右に移動しながら鑑賞するとき、モデル正面からおおよそ  $37^\circ$  の範囲であれば背面が見えないが、図 8(b) ではおおよそ  $59^\circ$  に広がる。反転基準の奥行きやスケーリング係数はともに、出力モデルの形状や、逆遠近錯視の立体が変形して見える度合いに影響する。これらはユーザが値を変更して調整できる。

第三段階では反転後の点をワールド座標系へ戻し、出力モデルとなる逆遠近錯視の立体を生成する。反転後のウィンドウ座標系の点  $\mathbf{p}'_w$  に対応するワールド座

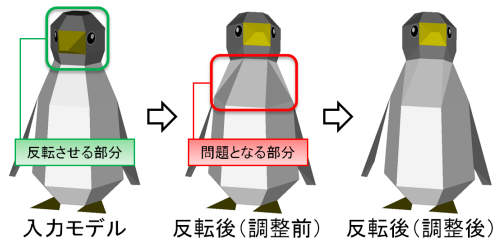


図 9: モデルの一部分だけを反転させる場合の流れ。

標系の点  $p'$  は次の式のようにになる。

$$p' = (VPM)^{-1}p_w \quad (3)$$

以上のように入力モデルを変形させることで、逆遠近錯視の立体を生成できる。得られるモデルは透視投影を考慮して反転されており、基準となる視点から見た場合、反転前と同じ形状であるかのように見える。なおテクスチャ画像については、一連の変形では入力モデルの頂点を動かしているだけでモデルの他の構造は変化しないので、入力モデルと同じテクスチャ画像を出力モデルにも使える。

### 3.2 指定部分に限定した凹凸反転の手法

生成できるモデルの表現の幅を広げるために、モデルを複数のパーツに分割しパーツごとに変形を行う。モデルをパーツ分割することで、図 1 のように逆遠近錯視が起こるパーツと通常形状のパーツを混在させることができる。さらに、パーツごとに反転基準とスケール係数を設定することで出力形状を調整することもできる。

本節では、はじめに、ユーザがパーツを指定し分割する方法について述べる。続いて、パーツごとに逆遠近錯視立体へ変形させる方法について述べ、最後に後処理として、パーツ間のつなぎ目領域を調整する方法について述べる。つなぎ目領域とは、通常形状パーツを構成している面のうち、逆遠近錯視パーツと接している面のことをいう。つなぎ目領域は通常形状パーツから逆遠近錯視パーツに遷移する部分であり、逆遠近錯視を妨げるような見た目の不自然さが生じる場合があるため、形状の調整が必要となる。モデルの一部を逆遠近錯視立体に変形させる場合の流れを図 9 に示す。

#### 3.2.1 パーツごとの凹凸の反転

パーツ単位で逆遠近錯視立体への変形を行うには、投げ縄ツールを用いてパーツを指定し、パーツごとに凹凸の反転を行う。図 10 にパーツ指定の例と、それによって選択された面を示す。ただし、単純にパーツごとに変形を行うと、パーツ同士が滑らかに繋がらな

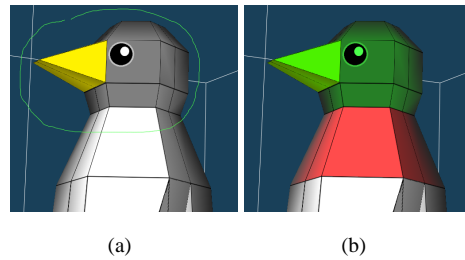


図 10: 投げ縄によるパーツ指定。マウス操作で線を描くと (a)、囲まれた領域内にある面が選択される (b)。図の緑色で塗られている部分が選択領域。赤色で塗られている部分がつなぎ目領域。

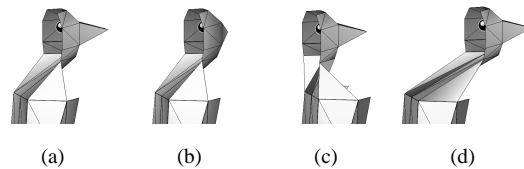


図 11: パーツごとに奥行きを変更。(a) 奥行きの変更なし、(b) 嘴の部分の奥行きを圧縮、(c) 頭部を手前に移動、(d) 頭部を奥へ移動。

くなくなってしまう。そこで、他のパーツと接している頂点群の中で基準視点から一番奥にある頂点の奥行きを、パーツの基準となる奥行き値  $z_p$  とし、反転基準  $z_b$  の代わりに用いることにする。すると、前節で示した奥行き反転の式 (2) が次のように変更される。

$$z' = z_p + (z_p - z) s \quad (4)$$

このパーツの基準値  $z_p$  および奥行きのスケール係数も、パーツごとにユーザが後から調整できる。図 11 に、パーツごとの奥行きの情報を変更した例を示す。

#### 3.2.2 つなぎ目領域の調整

通常形状パーツから逆遠近錯視パーツに移り変わるつなぎ目領域では、三角形メッシュの辺の接続性によって、入力モデルにはない起伏が生じる場合がある。この起伏によって物体の陰影が変化し、逆遠近錯視が起こりにくくなる、という問題がある。特に、本研究では粗いポリゴンモデルを対象としているため、陰影の変化が顕著となる。そこで、つなぎ目領域における三角形メッシュの辺の接続性を修正することで、この問題を解消する。陰影は面の法線に影響を受けるため、入力モデルから法線ができるだけ変化しないよう、辺のつなぎ替えを行う。図 12 は、辺のつなぎ替えの適用例である。

まず、つなぎ目領域の中から隣り合う二面を一組ずつ選び、辺をつなぎ替える前後で、それぞれ評価値  $E$  を計算する。この評価値は、入力形状と反転後形状の

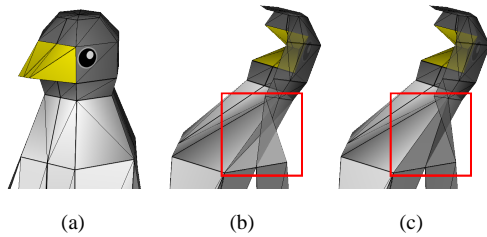


図 12: つなぎ目領域における辺のつなぎ替え. (a) 入力モデル, 辺のつなぎ替え前 (b) と後 (c) の出力モデル.

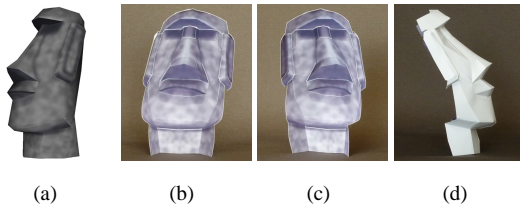


図 13: 提案手法の適用例 (モアイ像). (a) の入力モデルの全体を反転させると, (b)(c)(d) のような立体を生成できる. (b)(c) のように視点を変えて観察すると, 視点の位置に合わせてモアイ像が回転しているように見える. (d) のように横から見ると, 実際の凹凸がわかる.

面の法線方向が似ているほど大きくする. 三角形  $i$  とそれに隣接する三角形  $j$  について, 入力形状での単位法線ベクトルを  $\vec{n}_i, \vec{n}_j$ , 変形後の単位法線ベクトルを  $\vec{n}'_i, \vec{n}'_j$ , 変形後の面積を  $A'_i, A'_j$  とすると, 提案手法では, 評価値  $E$  は次の式で計算される.

$$E = \frac{A'_i(\vec{n}_i \cdot \vec{n}'_i) + A'_j(\vec{n}_j \cdot \vec{n}'_j)}{A'_i + A'_j} \quad (5)$$

ここで, より大きい面の影響が大きいと考え, 面積による重み付けを行なっている. また, 反転後の面積が重要と考え, 重み付けには反転後の面積を用いている. 面積の総和で正規化を行うのは, 評価値の比較を容易にするためである. 隣り合う二面のすべての組み合わせに対して評価値を計算した後, 評価値の大きい順に辺の接続性を決定していく. ただし, 一度確定された面はそれ以降の面の確定処理の候補から外す. また, 辺のつなぎ替えによってテクスチャ座標が不連続になるとモデルのテクスチャが不連続になってしまうため, そのようなつなぎ替えは予め処理候補から外す.

以上の処理により, 図 9 の調整前・調整後に示すように, つなぎ目領域にある陰影の急激な変化が目立たなくなる.

## 4 結果

提案手法を C++ 言語, OpenGL, GLUT, GLUI ライブラリを用いて実装した. 実行環境は Intel Core2 Duo

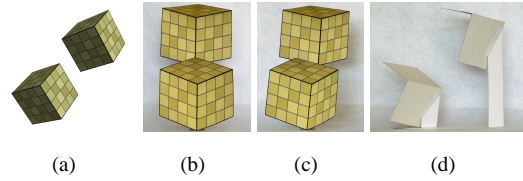


図 14: 提案手法の適用例 (二つのキューブ). (a) の二つのキューブを別々に指定して反転させ, (b)(c) のように視点を変えて観察すると, 視点の位置に合わせてキューブが回転しているように見える. (d) のように横から見ると, 実際の凹凸がわかる. 二つのキューブを別々に反転させているので, キューブ同士の前後の位置関係は保たれている.

CPU E6750 (2.66GHz), メインメモリ 2.0GB, GPU として NVIDIA GeForce 8600 GT を搭載した PC である. 提案手法を実装したシステムを用いると, ユーザはマウスで基準視点と反転部分の指定を行うだけで, 対話的な処理で逆遠近錯視立体を得られた.

本節では, はじめに提案システムによる適用例を示し, その後, 適用例を用いた被験者実験とその結果について述べる. 被験者実験では, ペーパークラフトの形で実世界に出力した逆遠近錯視立体を使用し, 実際に逆遠近錯視が起こるか, 人間の目で観察し調べた.

### 4.1 適用例

図 3, 図 13, 図 14 は, 提案手法による適用例である. それぞれ (a) が入力に用いた三次元ポリゴンモデル, (b)(c)(d) が生成されたモデルをペーパークラフトにしたものである. 市販のソフトウェア<sup>1</sup>を使って展開図を生成し, 組み立てた. 各ペーパークラフトの高さは 20cm 程度で, 透視投影に用いた垂直視野角は 30° とした. 基準視点はいずれの適用例も (a) の画像の左方向にある.

図 13 のモアイ像はモデル全体が反転されており, 図 3 のフクロウは頭部だけが反転されている. どちらの例も, 視点の位置を変えても常に顔が観察者の方向を向いているように見える.

図 14 の適用例では, 二つのキューブを別々に反転させているため, キューブ同士の前後の位置関係が保たれている. もし二つのキューブを一緒に反転させると, 図 15 に示すように前後の位置関係が入れ替わってしまうが, ユーザが反転の基準となる奥行きをキューブごとに指定することで, この問題を解消できる.

### 4.2 被験者実験

提案手法で生成した立体が本当に逆遠近錯視を引き起こすかどうか確認するために, 生成したモデルのペーパークラフトを用いて被験者実験を行った. コンピュータ上ではなく実世界のペーパークラフトとして

<sup>1</sup> 『ペーパーデザイナー 3』(多摩ソフトウェア)

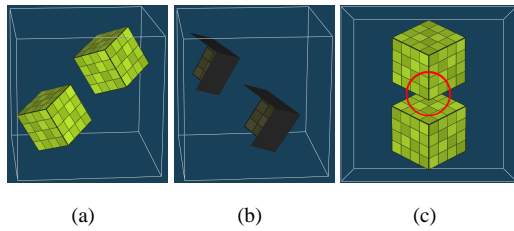


図 15: 二つのキューブを一緒に反転させた例。(a)の二つのキューブを一緒に反転させると(b)のようになる。基準視点は画像左方向にある。基準視点から見ると(c)のようになっており、キューブ同士の前後の位置関係が入れ替わってしまっている。

逆遠近錯視立体を鑑賞する場合、3.2.2節で触れた通り、陰影が凹凸の認識に影響し、逆遠近錯視が起きにくくなることもある。そこで、照明環境を変えて陰影の影響についても調査した。

#### 4.2.1 実験方法

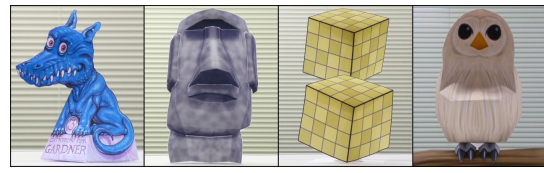
6名の被験者に、基準視点と同じようにモデルが見える位置に立ってもらい、片目を瞑った状態で視点を動かしたときにモデルがどのように見えるか答えてもらった。基準視点は正面方向に2メートル離れた位置に設定した。片目で観察する理由は、両目で観察すると両眼視差によって本当の凹凸がわかりやすくなってしまからである。逆遠近錯視は、凹凸を誤って認識することで起こる錯視なので、本当の凹凸ができるだけわからない状態で観察することが望ましい。陰影が凹凸の認識に与える影響を調べるため、光源が天井の蛍光灯のみの場合と、モデルを正面からライトで照らして陰影を目立たなくした場合の二種類の方法で実験を行い、それぞれの場合についてモデルがどのように見えるか答えてもらった。被験者のうち5名は逆遠近錯視が起きる仕組みを知っており、残りの1名は逆遠近錯視に関する知識を持っていなかった。実験にはモアイ像、フクロウ、二つのキューブを使用し、さらに比較のために図1のドラゴンも使用した。図16に、実験に用いた逆遠近錯視立体を示す。

#### 4.2.2 実験結果と考察

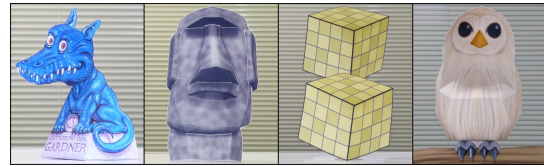
実験の結果、次のような結果が得られた。

ドラゴン 「頭部だけが視点に合わせて回転して見える」という回答が全員から得られた。また、正面から照らすライトの有無に関わらず、逆遠近錯視が起きた。

モアイ像 「モデルの全体が視点に合わせて回転しているように見える」という回答が全員から得られた。しかし、正面から照らすライトに関しては「ライトの有無による見え方の変化はない」と答えた被験者が4人、「ライトがあるほうが錯視が起こりやすい」と答えた被験者が1人、



(a) 正面から照らすライトがオフの状態



(b) 正面から照らすライトがオンの状態

図 16: 実験対象の逆遠近錯視立体。左からドラゴン、モアイ像、キューブ、フクロウ。(a) 正面から照らすライトがオフの状態に比べ、(b) オンの状態のほうが陰影が少ない。陰影は本当の凹凸を知る手がかりになり得るため、正面から照らすライトの有無による見え方の違いについても調べた。

「ライトがないほうが錯視が起こりやすい」と答えた被験者が1人いた。

二つのキューブ 「どちらのキューブも視点に合わせて回転しているように見える」という回答が全員から得られた。しかし、正面から照らすライトに関しては「ライトの有無による見え方の変化はない」と答えた被験者が3人、「ライトがあるほうが錯視が起こりやすい」と答えた被験者が2人、「ライトがあるときに限り錯視が起こる」と答えた被験者が1人いた。

フクロウ 「顔部だけが視点に合わせて回転して見える」という回答が全員から得られた。しかし、正面から照らすライトに関しては「ライトがあるときに限り錯視が起こる」と答えた被験者が5人、「ライトがないほうが錯視が起こりやすい」と答えた被験者が1人いた。さらに前者の中から、「頭部と胴体が繋がっていないように見える」という意見が2人から得られた。

以上の結果から、提案手法を用いて生成されたモデルを観察すると、実際に逆遠近錯視が起こることがわかった。特に、ドラゴン・モアイ像・フクロウのように顔や目が付いたモデルでは、回転しているように見えるだけでなく、常に視線が追跡されているように感じることがわかった。一方で、陰影の影響で錯視が起こりにくくなる場合があることがわかった。

## 5 結論と今後の課題

本研究では、入力された三次元ポリゴンモデルの奥行きを投影変換に基づいて反転することで、逆遠近錯視立体を生成する手法を提案した。また、生成できる

モデルの表現の幅を広げるために、ユーザがモデルの一部分を指定することで、指定した部分だけを逆遠近錯視立体へ変形させることも可能とした。

さらに、出力結果をペーパークラフトにしたものを用いて被験者実験を行い、実際に錯視が起こることを確認した。被験者実験の結果、陰影の影響で錯視が起こりにくくなる場合があることがわかった。陰影の影響を減らすためには、物体の色を濃くする、CGで陰影や影を計算した入力モデルの画像をテクスチャとして用いる、つなぎ目領域の調整方法を改良する、などが考えられる。

また、提案手法では可視判定を面単位で行っているため、本来不要な面の一部だけが見えていても、その面が削除されずに残ってしまうことがある。これを解決するには、投影されたモデルの輪郭に沿って面を再分割するなどして、可視判定をより精密に行う必要がある。さらに、反転の基準視点から見て入力モデルの奥行きに大きな段差がある場合、生成される逆遠近錯視立体にも段差が生じてしまい、鑑賞できる角度が著しく制限される。例えば同じモデルの奥の部分が手前の部分に隠されながら一部だけ見えている、という場合、遮蔽の境界で段差が生じる。このようなモデルへの対応は今後の課題のひとつである。

他に、ペーパークラフトに限らず 3D プリンタを使った実世界への出力を想定し、より複雑な形状のモデルに対して適用させるための発展などが考えられる。

## 参考文献

- [1] Cook, N.D., Hayashi, T., Amemiya, T., Suzuki, K., and Leumann, L.: Effects of visual-field inversions on the reverse-perspective illusion. *Perception*, Volume 31, Issue 9 (2002), pp. 1147–1151.
- [2] Cook, N.D., Yutsudo, A., Fujimoto, N., and Murata, M.: Factors Contributing to Depth Perception: Behavioral studies on the reverse perspective illusion. *Spatial Vision*, Volume 21, Numbers 3-5 (2008), pp. 397–405.
- [3] Cook, N.D., Yutsudo, A., Fujimoto, N., and Murata, M.: On the visual cues contributing to pictorial depth perception. *Empirical Studies of the Arts*, Volume 26, Number 1 (2008), pp. 67–90.
- [4] Grand Illusions: Dragon Illusion. [http://www.grand-illusions.com/opticalillusions/dragon\\_illusion/](http://www.grand-illusions.com/opticalillusions/dragon_illusion/).
- [5] Hayashi, T., Umeda, C., and Cook, N.D.: An fMRI study of the reverse perspective illusion. *Brain Research*, Volume 1163 (2007), pp. 72–78.
- [6] Hughes, P.: reverspective.com. <http://www.patrickhughes.co.uk/>.
- [7] OpenGL 策定委員会 (松田晃一 訳): 『OpenGL プログラミングガイド 原著第 5 版』. ピアソン・エデュケーション, 東京, 2006.
- [8] Owada, S., and Fujiki, J.: DynaFusion: A Modeling System for Interactive Impossible Objects. *Non-Photorealistic Animation and Rendering (NPAR) 2008*, pp. 65–68.
- [9] Papathomas, T.V.: Experiments on the role of painted cues in Hughes’s reverspectives. *Perception*, Volume 31, Issue 5 (2002), pp. 521–530.
- [10] Sugihara, K.: Three-dimensional realization of anomalous pictures —An application of picture interpretation theory to toy design. *Pattern Recognition*, Volume 30, Issue 7 (1997), pp. 1061–1067.
- [11] 杉原厚吉: 『立体イリュージョンの数理』. 共立出版, 東京, 2006.
- [12] 高野もも: 「逆遠近錯視立体の自動生成に関する研究」. 東京工科大学メディア学部, 2008 年度卒業論文.
- [13] Wu, T.-P., Fu, C.-W., Yeung, S.-K., Jia, J., and Tang, C.-K.: Modeling and Rendering of Impossible Figures. *ACM Transactions on Graphics (ACM TOG)*, vol. 29, No. 2, Article 13 (March 2010).

伊藤 紘治 (いとう ひろはる)



2011 年筑波大学情報学群情報科学類卒業 . 同年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程入学 . コンピュータグラフィックスの研究に興味を持つ .



金森 由博 (かなもり よしひろ)



平成 21 年 3 月東京大学情報理工学系研究科コンピュータ科学専攻博士課程修了。博士（情報理工学）。同年 4 月より筑波大学に勤務し、現職は筑波大学システム情報系・助教。リアルタイムレンダリング、ビジュアルシミュレーションや画像編集技術に興味を持つ。

三谷 純 (みたに じゅん)



筑波大学システム情報工学研究科准教授。2004 年東京大学大学院工学系研究科博士課程修了。博士号取得（工学）。理化学研究所研究員，筑波大学講師を経て 2009 年より現職。主にコンピュータグラフィックスおよび折紙の研究に従事。

福井 幸男 (ふくい ゆきお)



筑波大学システム情報系教授。1980 年東京大学大学院工学系研究科機械工学修士課程修了。同年工業技術院製品科学研究所入所。同生命工学工業技術研究所を経て、1998 年筑波大学電子・情報工学系に移籍。2004 年組織変更で現職。