

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

吉川祐輔[†](非会員) 宮下芳明^{†, ‡}(正会員)

[†] 明治大学大学院 [‡] 独立行政法人科学技術振興機構, CREST

Creation Techniques of Illusion Expression in Real-time by Combining Image Buffers

Yuusuke Kikkawa[†] Homei Miyashita^{†, ‡}

[†] Meiji University [‡] JST, CREST

概要

人は、現実には存在し得ないものをイメージすることができる。例えば、エッシャーの絵画に描かれているものに代表される「不可能立体」と呼ばれる立体は、脳内ではそれがあたかも存在する立体のように感じられるが、現実には作ることにはできない。「不可能立体」のような現実には存在し得ない立体が表現可能になれば、それは人の創造の可能性を広げるはずである。本論文では、複数の画像バッファをリアルタイムに合成し錯覚表現を作り出す手法を提案し、そのアルゴリズムと作例について述べる。提案手法を用いることで、現実には存在し得ない不可能立体を様々な角度から観察することや、時空間を歪ませる立体がそこにあるかのような表現が可能になる。本手法では、画素情報を用いて画像を組み合わせるといった平面的な処理しか行っていないにもかかわらず、あたかも立体的な変化を起こしているかのような表現が得られる。

Abstract

We can image some objects that cannot exist. For example, M.C. Escher drew some “impossible objects” in his works, which we can see as “real objects” despite those cannot be created in real. If we can express imaginable objects as impossible objects, that will extend our creation potential. In this thesis, we propose techniques that create illusion expression by combining several image buffers in real time, and describe those algorithms and sample works. Using those techniques, it is able to show impossible objects as real objects and depict temporal-spatial distortion objects. Those methods are planar process that only combining images using pixel information. Nevertheless, we obtain expressions like using depth information.

1 はじめに

ARを用いて現実環境上に仮想物体を3DCGとして重畳するとき、現実環境と仮想物体の間で整合性を取ることは大きな課題であり、それらの解決によってより現実味の増したAR表現が可能となる[1]。だが一方で、現実と区別できないような仮想物体を描画することがAR技術の最終到達点とは必ずしも言えないだろう。写真の登場によって絵画表現が廃れることが無かったのと同様に、ARを人の表現技法の一つとして捉えるならば、むしろその技術は通過点にすぎない。

人は、現実存在し得ないものをイメージすることができる。例えば、エッシャーの絵画に描かれているものに代表される、脳内ではそれがあたかも存在する立体のように感じられるが、現実には(少なくともそれを見て脳が感じたように)作ることのできない立体は「不可能立体」あるいは「不可能物体」と呼ばれる[2]。ARによって存在「しない」ものを存在しているように見せられる今「不可能立体」のような現実には存在「し得ない」立体が表現可能になれば、それは人が想像可能なイメージをその通りに伝える表現の可能性を広げるはずである。

著者はこれまで、AR上で不可能立体を提示する手法について提案し[3]、またその手法を応用することで、不可能立体に限らない様々な錯覚表現を可能とした[4]。本論文では、さらにそれらを発展させ、更に多数の錯覚表現の生成を可能にした。これらをARアプリケーション上で実装することで、それらがあたかもそこにあるかのように観察できる。本論文で提案する手法を用いて作られた作例は、[5]の動画を参考にされた。

本論文で提案する錯覚表現生成手法は全て、「カメラから取得された画像バッファ」と「その画像を入力とし、ARオブジェクトが描かれた画像バッファ」を組み合わせたものである。この手法は、ARの描画結果を待たなければならないため、全体的な処理負荷が大きいという欠点がある。だが一方で、「ARオブジェクトがの座標系の計算」及び「ARオブジェクトの描画処理」に一切依存せず、ARを描画することが可能な環境ならば利用可能である、という大きな利点がある。また、本論文で示す作例では、提案手法を主にARに適用しているが、ARに限らず単に3DCGを描画可能な環境にも転用可能である。

なお「錯覚」という言葉は「外界の事物をその客観的性質に相応しないで知覚すること」を指して用いられることが多い[6]。本論文では「錯覚表現」という言葉を「本当のものとは違ったものを表現すること」という意味で用いる。従って、本論文で取り扱うのは、静止画なのに動いて見えたり同じ長さの線なのに違った長さに見えたりするような、脳が受け取った情報を誤って解釈してしまう「内的な錯覚」ではなく、コップに挿したストローが曲って見えたり屋敷楼で遠くのもの水の上に浮かんで見えたりするような、与えられる刺激としては正しいけれども実際とは違う「外的な錯覚」を与える表現である。

以下本論文では、第2節から第5節にかけ、著者が提案する種々の表現手法についてそれぞれの節で個別に述べる。そして、第6節にてそれらのまとめに

ついて述べる。

2 不可能立体の表現

脳内でそれがあたかも存在する立体かのように感じられるが、現実には(少なくともそれを見て脳が感じたように)作ることのできない立体は「不可能立体」と呼ばれる。代表的な不可能立体には、ペンローズの三角形やペンローズの階段、悪魔のフォークなどがある。

不可能立体は、平面に描かれたものであってもその奥行きを自動的に補完する知覚を利用した錯覚表現である。そのため、その表現は平面的描画に留まらざるを得ない。杉原は、不可能立体が本当に実現不可能であるのかについて言及し、特定の角度から見ることで不可能立体として観察可能な立体が存在することを示し、実際にそれらを実物体として製作した[7]。それらは、これまで平面的描画でしか表現できなかった不可能立体を、実物体として表現するという新たな可能性を示した。

一方、本論文で示す手法では、遮蔽関係に矛盾が生じるタイプの不可能立体をリアルタイムに描画することができる。これにより、不可能立体を静的な一枚絵としてではなく、ある程度自由な角度から動的に観察することを可能とした。本手法は平面的描画に留まっただけのもの、様々な角度から観察可能であるという点で、杉原の手法とは違った意味で不可能立体の存在感を表現する事が可能である。

2.1 関連研究

不可能立体を計算機上で表現しようとする試みには様々なものがあり、それらには、立体をワイヤフレームモデルのように辺だけを表現するものと、サーフィスモデルのように面を描画することで表現するものがある。ここでは、前者を「線画的表現」、後者を「面画的表現」と呼ぶことにする。

Chihらは不可能立体を実現可能な部分立体の接続によって表現し、線画によって表された不可能立体の回転アニメーションの生成手法を提案している[8][9]。これによって生成されたアニメーションは、回転しながらもその構造を保持し続ける一方で、その構造の保持のために立体感は犠牲になっている。大和田らは、線画によって表された三次元の可能立体の、不要な辺の削除と任意の辺同士接続によって、線画的表現の不可能立体をインタラクティブにモデリングできるツールを開発した[10]。これによって作られた不可能立体は、視覚的連続性を失うことなく回転させることができる。藤木らは、インタラクティブなだまし絵やそのエンタテインメント性を発展させたゲームを提案している[11][12]。これは、三次元オブジェクトをディスプレイ上で表すとき一つの側面からしか観察できないことを用い、ディスプレイ上での見え方によってキャラクタの振る舞いを変えようというものであり、実世界ではありえないようなキャラクタの徘徊操作が

行える。これら「線画的表現」では、「面画的表現」ではできない不可能立体を表現可能な部分もあるが、写実的表現には向かない。

Savransky らは不可能立体を構成する部分ごとに各画素の深度情報を持った画像のレンダリングを行い、その深度情報を元にユーザが指定した遮蔽関係グラフを用いて不可能立体を描画する手法を取った [13]。第二著者らは、不可能立体のレイトレーシングを行う手法を提案した [14][15]。第二著者らの手法では、「奥行き逆転」と「面の不自然な接合」のトリックを使った不可能立体を描画することが可能である。「奥行き逆転」は、レイトレーシングを行う際、本来レイが当たるオブジェクトよりも後方にあるオブジェクトにレイを当てることで、背面にあるオブジェクトが前にあるようにレンダリングされる、というものであり、本論文の手法の考え方の元となった「面の不自然な接合」は、まず線画的表現のように立体の辺を繋いで不可能立体を作り、それによって現れる面をレンダリングすることができる手法である。Wu らの不可能立体のレンダリング手法 [16] では、不可能な面の接合をカメラ位置から見たときに自然に見えるようにするアルゴリズムを用いている。これは第二著者らの「面の不自然な接合」に類似しているが、第二著者らの手法は、「面があるように見える部分に対し面があるようにレンダリングする」のに対し、Wu らの手法は「カメラ位置から自然に見えるように実際に面をねじ曲げる」という違いがある。McCann らのシステム [17] は、通常のペイントソフトにおけるレイヤ機能を拡張し、レイヤ同士が重なっている部分に対してそれらのレイヤを分割しその部分のスタックの順番を任意に入れ替えることのできるシステムである。これを用いて、遮蔽関係が入れ替わった不可能立体のレンダリングを行うこともできる。

本論文の手法も、これら「面画的表現」に含まれる。本手法は、三次元的な計算を行わず立体の構造自体には手を加えないので、作れる不可能立体の種類は限られるものの、その一方で、3DCG さえ描画できれば応用可能なので環境的な汎用性は高い。

2.3 節では鏡を用いた作例を示すが、鏡を用いたエンタテインメントシステムとしては、禹らの Heaven's Mirror [18] や箕らの through the looking glass [19] などがある。Heaven's Mirror は、鏡の中の物体の見え方によって実物体を動かしたり、鏡を V 字に組み合わせさせて鏡像がいくつにも増えて見えるとき実際に物が増えたかのように複数音源を提示したりすることで、鏡が作り出す錯覚を増幅させる作品である。through the looking glass は、特定の角度範囲からの光だけを拡散させそれ以外の角度からの光を透過する特殊な建材を使ったスクリーンを用いて、現実世界のスクリーンと（それが写っているはずの）鏡の中のスクリーンに違う映像を投影することで、本物の鏡を用いながら鏡の中に別の世界を映し出す作品である。

2.2 アルゴリズム

ここでは、アルゴリズムの概念図 (図 1) を用い、U の字型と I の字型のオブジェクトを組み合わせた不可

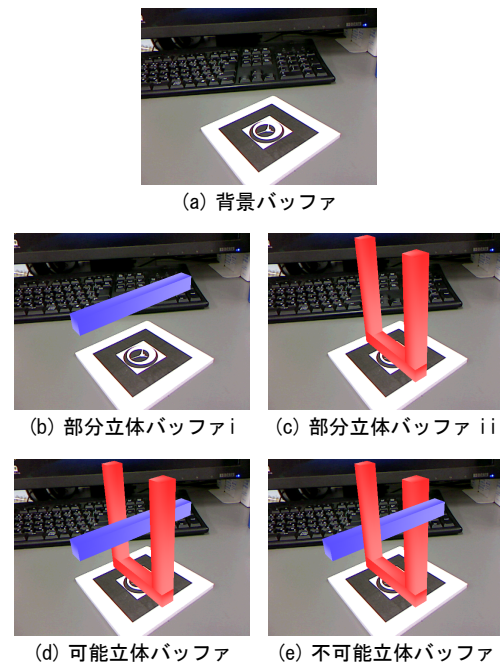


図 2: 不可能立体 AR の画像バッファ

能立体表現と画像バッファ群 (図 2) を例に説明する。

「背景バッファ」は、ウェブカメラでキャプチャした画像が格納されているバッファである (図 2(a))。「部分立体バッファ」とは、不可能立体を構成するオブジェクトの一部 (部分立体) が描画された結果が格納されたバッファである。図 2 においては、U の字型のオブジェクトと I の字型のオブジェクトがそれぞれ部分立体であり、それらが描画されたバッファ (図 2(b), (c)) が「部分立体バッファ」である。「可能立体バッファ」とは、不可能立体を構成するオブジェクトを通常の遮蔽関係で描画した結果が格納されたバッファである (図 2(d))。そして、それらのバッファを組み合わせることで不可能立体が描画された結果が格納されるのが「不可能立体バッファ」である (図 2(e))。

まず始めに、カメラで撮影した画像を「背景バッファ」に格納する。そしてその画像を元に部分立体と可能立体を描画し、それぞれを「部分立体バッファ」と「可能立体バッファ」に格納する。「部分立体バッファ i, ii」は図 1 の A, B, 「可能立体バッファ」は図 1 の C に当たる。以降、A に描画される部分立体を A , B に描画される部分立体を B と呼ぶ。

次に「部分立体バッファ」と「背景バッファ」の各画素を比較し、部分立体が描画された領域 (図 1 の) を取得する。これは、背景差分法によって各部分立体が描画された領域を毎フレーム取得しているのに等しい。

と の集合積を取ることで、部分立体がスクリーン上で重なっている領域 (図 1 の) が取得できる。この領域 に含まれる画素に対して、その画素の色を「部分立体バッファ i, ii」と「可能立体バッファ」から

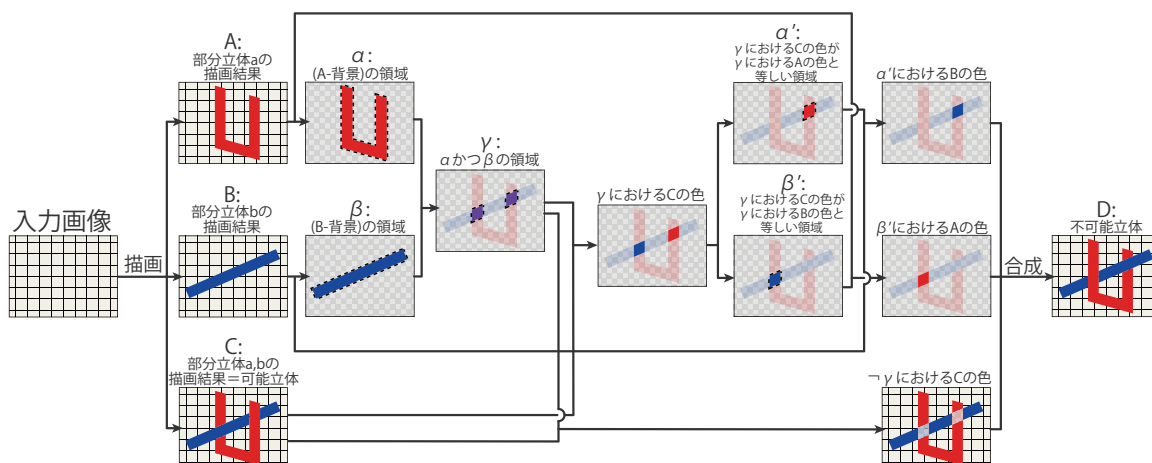


図 1: 不可能立体描画のアルゴリズム概念図

それぞれ取得し、それらの色を比較することで『スクリーン上でどの部分立体が前にあるのか』が分かる。例えば、『可能立体バッファ』から取った色が『部分立体バッファ i 』から取った色と等しかったならば、『スクリーン上では部分立体 A が前にある』ことが分かる。この領域を α と置く。また、『部分立体バッファ ii 』から取った色と等しかったならば、その領域を β と置く。この時、領域 γ は領域 α と領域 β の直和である。

『不可能立体バッファ』(図 1 の D) は、『部分立体バッファ i, ii 』、『可能立体バッファ』C を組み合わせて生成する。まず、領域 $\neg\gamma$ (領域 γ 以外の領域) の画素においては『可能立体バッファ』から色を取得する。『不可能立体バッファ』D における各画素において、その画素が含まれている領域が領域 α であればその画素の色は『部分立体バッファ i 』から取得し、その画素が含まれている領域が領域 β であればその画素の色は『部分立体バッファ ii 』から取得する。

これにより、『不可能立体バッファ』D の領域 α において、本来後ろにあるオブジェクトの色がスクリーン上に現れることになり、その結果オブジェクトの遮蔽関係が入れ替わったかのような画像出力が得られる。

本手法を用いる上では、オブジェクトの描画にアンチエイリアス処理を掛けてはいけないという注意点がある。オブジェクトの描画にアンチエイリアス処理を掛けると、オブジェクトのエッジ部の色に変化してしまうことがある。本手法では画像バッファの画素の色を利用しているため、アンチエイリアス処理により部分立体の描画領域がうまく抽出できなくなり、図 3 の様に部分立体のエッジが残りに不自然な描画になってしまう場合がある。

本手法を用いることで、その画素においてどのオブジェクトが最もスクリーンに近いのかを、デプスバッファの値などにアクセスすることなく取得できる。もちろん、3D オブジェクトの遮蔽関係に従って描画するための計算処理は内部で行われているが、このアルゴリズムではそれに関与することなく、描画された結

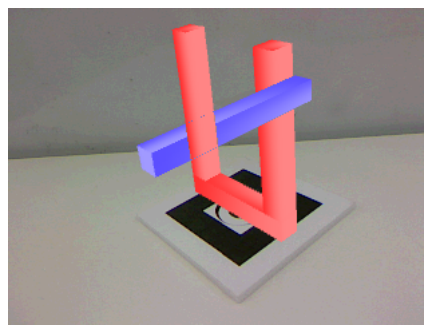


図 3: エッジが残る描画

果だけを用いて利用可能である。そのため、3DCG のレンダリングができる環境であれば、同様のアルゴリズムを用いることで不可能立体を描画することが可能である。例えば著者は同アルゴリズムを用いて、ウェブ上で不可能立体を閲覧可能な Java Applet を作成した(不可能な交差¹、ネッカーキューブ²)。

2.3 作例

2.2 節のアルゴリズムを用いることで、二つのオブジェクトの遮蔽関係が逆転する不可能立体が描画でき、その例が図 4(a) である。また、このアルゴリズムを応用することで、二つのオブジェクトが重なった領域の遮蔽関係を『逆転』ではなく『固定』することもできる。遮蔽関係の固定とは、例えば部分立体 A と部分立体 B が重なった領域に対して、常に部分立体 B が手前にあるように描画することを指す。

¹<http://miyashita.com/ImpossibleObjects/index.html>

²<http://miyashita.com/NeckerCube/index.html>

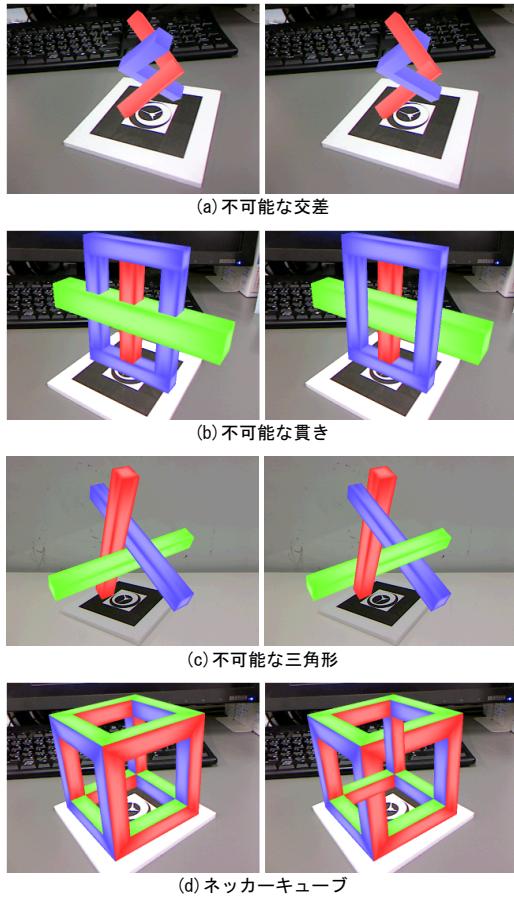


図 4: 不可能立体 AR の作例 左: 可能立体 右: 不可能立体

図 4(b), (c), (d) は、遮蔽関係を固定することで描画した不可能立体である。図 4(b) では、緑色のオブジェクトは、常に青色のオブジェクトの背面かつ赤色のオブジェクトの前面に描画される。図 4(c) では、赤色のオブジェクトは常に緑色のオブジェクトの前面に、緑色のオブジェクトは常に青色のオブジェクトの前面に、青色のオブジェクトは常に赤色のオブジェクトの前面にそれぞれ描画される。図 4(d) では、立方体を構成する辺を三つの部分立体に分け、それらの遮蔽関係を固定することで描画している。

図 5 は、元のマーカ上には「可能立体」を、鏡に写ったマーカ上には「不可能立体」を描画するようにしたものである。これによって、現実世界と鏡の中の世界で同じ形状の可能立体と不可能立体を対比させて見ることができ、より恣意的な観察を可能とする。これは、鏡に写ったマーカは元のマーカと線対称な別のマーカとして認識できることを利用している。

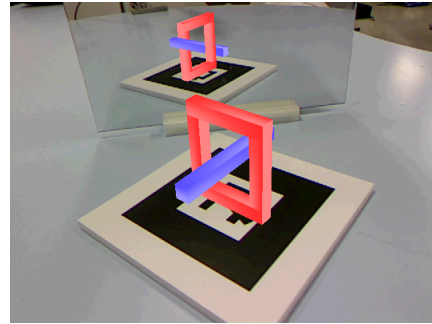


図 5: 鏡の中に映った不可能立体

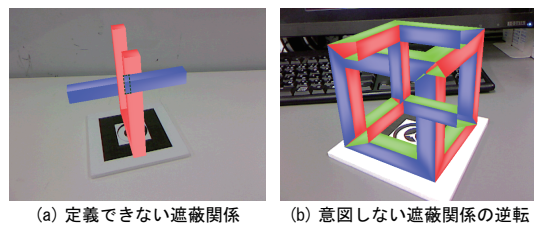


図 6: 不可能立体の破綻

2.4 今後の課題

不可能立体を様々な角度から観察する上で発生する問題として、観察する角度によって不可能立体が「破綻してしまう」ことが挙げられる。例えば図 6(a) では、青い I の字型のオブジェクトが、赤い U の字型のオブジェクトの後ろの棒を貫通しているかのような不自然な描画となってしまっている。これは、図 6(a) の破綻で囲った内側の画素が「赤(後ろ)-青-赤(前)」という三重の遮蔽関係を持っているからである。この画素の場合、赤い U の字型のオブジェクトと青い I の字型のオブジェクトの遮蔽関係を単純に入れ替えることができない。

Savransky らの不可能立体の描画手法においても、三つのオブジェクトの遮蔽関係が一意に定義できない領域について、その領域はレンダリングすることが不可能であると述べている [13]。Wu らは、自らのシステムにおいて、不可能図形の視点範囲を変えたときにどの程度誤差が発生するかを調べ、それをを用いることであまりにも大きな破綻が起きないようにできると述べている [16]。しかしながら本システムでは、マーカとカメラの位置関係を制限する手段が無いため、不可能立体を観察する際「破綻が起きない角度からだけ観察させる」ことはできない。

また、本手法を用いて描画した不可能立体を裏側から見たとき、図 6(b) のように意図していない部分の遮蔽関係が逆転してしまうことがある。この問題は、マーカとカメラの位置関係から、遮蔽関係の逆転をするかしないかを分岐させることで解決可能であると思われる。しかしながら、裏側から見て破綻するような

不可能立体の場合、それを裏側から見ようとした時、その破綻が起きる前に前述の三つ以上のオブジェクトが重なることによる破綻が起きてしまうため、根本的な解決には至らない。

これらのように、ある角度から観察することで不可能立体が破綻する問題については、表現上の問題として今後解決すべき課題である。

また本手法では、不可能立体を作る前に、幾つかの事前準備が必要である。まず一つ目に、不可能立体の元となる「可能立体」を「部分立体」に分割しておく必要がある。二つ目に、それぞれの遮蔽関係の入れ替え方を、出来上がりを頭の中で想像して定義しておかなければならない。そして、それらの分割や遮蔽関係が上手く作用するのは、不可能立体を描画してみることで初めて分かるものとなっている。そのため、物体の位置の微調整には手間がかかり、遮蔽関係の設定も難しく、不可能立体の制作環境としては大きな改善余地がある。

Igarashi らは、布やひもなどの折り重なった柔軟物体の重なり順を、クリックや物体ドラッグ時のシフト操作で指定するインタラクション手法を提案している [20]。その手法のように、動的に遮蔽関係を入れ替えることができ、それによって重なっている部分に矛盾が発生しないようにすることができれば、より簡単に不可能立体を作成することが可能になるだろう。

3 時空間を歪める表現

本章で対象とする「時空間を歪める表現」は、カメラが映し出す投影面の一部を切り取り、そこに数秒前の過去を重ねることで、“同時”に“異なる時間軸座標”から物事を観察することができるものである。単に電子情報を付加する AR とは違い、この表現は現在しか映さないはずの投影面の一部で時間軸をずらして見せることができる、新たな現実感の拡張である。

3.1 関連研究

高嶋らは、過去に撮った映像の時間の流れ方と見せ方をユーザが自由にコントロールできる可視化表現として TbVP(Time-based Visual Presentaion) という概念を提唱し、またそれに基づき TbVP Browser を構築した [21]。TbVP Browser は、映像の一部領域に対して映像の速度をユーザが任意に変化させることで、周りとは異なる時間が流れるような表現をユーザに提示することができる。

Khronos projector[22][23] は、映像が投影されたスクリーンに触れることで、スクリーンを触ってたわませるように映像の時間軸をたわませることができるアートインスタレーションである。また、Snail Light Projector[24] は、プロジェクタから超低速の光がスクリーンに投影されているかのように、スクリーンとプロジェクタの距離によって投影される映像の時間軸座標を定めるシステムである。

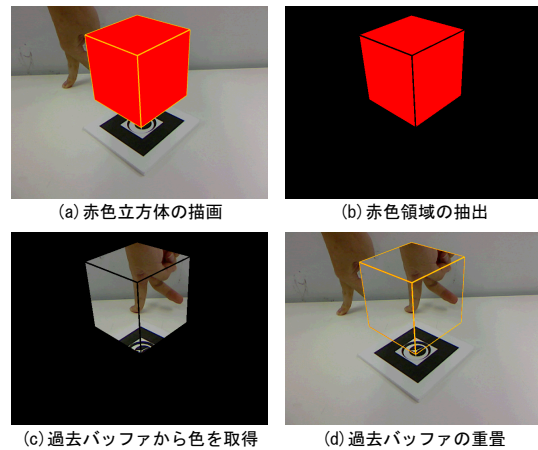


図 8: 時空間を歪める表現の画像バッファ

ドアコム [25] は、二つのカメラ入力画像を用い、片方ではドア型のデバイスを掲げその中に顔を映し、そのドア型デバイスの領域をもう片方のカメラ画像に合成することで、遠隔二地点でビデオチャットを行うシステムである。画像合成により空間を繋げて見せることができるドアコムは AR 表現的な側面も持っており、本論文でこれから示す手法ともよく似ている。本手法においても、ユーザによって定義された一部領域に対して別の画像を合成することができ、それを利用して時間や空間を超越した映像の提示を行う使用法が考えられる。

3.2 アルゴリズム

以下、本論文ではディスプレイ上の色を (R:r, G:g, B:b) という表記をもって示す。R, G, B は赤 (Red), 緑 (Green), 青 (Blue) の頭文字である。r, g, b は色の赤要素、緑要素、青要素の成分を示す 0 以上 255 以下の値を取る整数であり、値が大きいほどその成分が強いことを示す。

本アルゴリズムの概要は図 7 に示す。まず過去フレームを格納する画像バッファにカメラから取得された現在の入力画像を格納する。以降、この過去フレームが格納された画像バッファを「過去バッファ」と呼ぶ。次に、通常の ARToolKit を用いてオブジェクトを描画するのと同様に、入力画像からマーカ座標系を求め、マーカ上に赤色 (R:255, G:0, B:0) の立方体を描画する (図 8(a))。以降、本論文ではこの赤色の立方体が描画されたバッファ上の領域を「キューブ領域」と呼ぶ。そしてその後、全ての画素を走査し、各画素の色を取得する。その画素の色が (R:255, G:0, B:0) ならば (図 8(b))、過去バッファから同座標の画素の色を取得し (図 8(c))、キューブ領域に重畳描画する (図 8(d))。カメラで撮影した映像上からは (R:255, G:0, B:0) という極端な赤色は現実的にほとんど取得されないため、この手法でもキューブ領域を簡易に取得することができる。

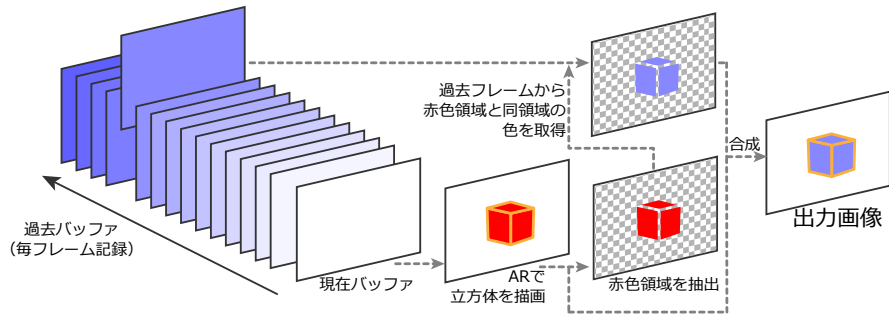


図 7: 時空間を歪める表現のアルゴリズム概念図

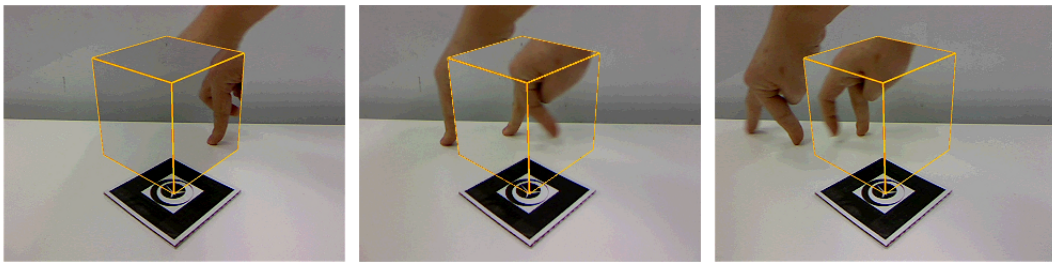


図 9: 時空間を歪める表現の使用例

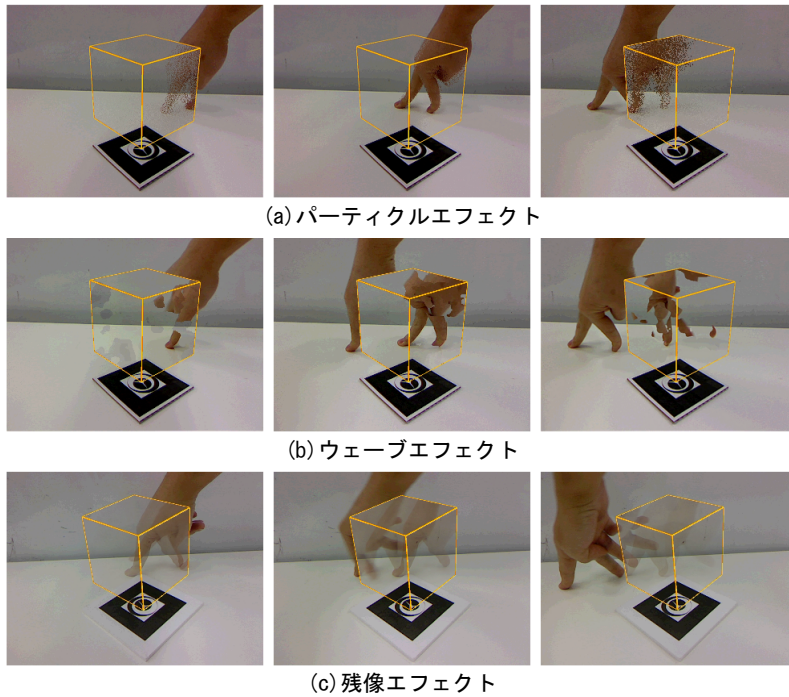


図 10: 様々なエフェクト

これにより、カメラを固定して移動する物体を映し出すと、キューブ領域だけ時間が遅延したような錯覚表現を得ることができる(図9)。また、カメラを移動しながら使用すると、キューブ領域の空間が歪んだような表現を得ることもできる。

3.3 エフェクト

キューブ領域に対して上書きするとき、単に一定時間前の過去バッファの画素を重畳するのではなく、どのバッファの画素を用いるのかを変えることによって様々なエフェクトをかけることができる。

パーティクルエフェクト(図10(a))は、キューブ領域のそれぞれの画素に対して、乱数を用いたランダムな過去バッファの画素を重畳するものである。これによって、キューブ領域内を動く物体が粒子化したかの様な効果が得られる。

ウェーブエフェクト(図10(b))は、パーティクルエフェクトと同様に乱数によって用いる過去バッファを決定するが、乱数の生成に(x座標, y座標, 経過フレーム数)を軸とする三次元のパーリンノイズを用いる。このx座標とy座標は、画素の画面上での座標を指す。パーリンノイズによって生成される乱数は、その次元空間上での距離が短いほど近い値を取る。そのため、隣り合った画素に重畳されるのは近い過去バッファのものとなり、またそれは時間経過によって徐々に変化していく。その結果、キューブ領域内を動く物体が波打っているかのように見えるエフェクトがかかる。

残像エフェクト(図10(c))は、現フレームから数フレーム前までの過去バッファを合成することで、残像のように見えるものである。現フレームに近いほど濃く見えるような重み付けをして合成することで、過去に映ったものが徐々に薄れていくように見えるようにした。

3.4 考察

福地らが制作したEffecTV[26][27]は、リアルタイムに入力される画像の動きを反映した映像効果を与えるビデオエフェクトソフトウェアである。本手法での「過去バッファ」は、福地らが「タイムバッファ」と呼んでいる手法を参考に、リングバッファに最近数十フレーム分の画像を置きそれを利用している。また、3.3節で挙げたパーティクルエフェクト(図10(a))の処理は、EffecTVのエフェクトの一つであるQuarkTVと同様の手法を用いている。他にも、本システムでのエフェクトはEffecTVで提供されている幾つかのエフェクトと類似しており、ウェーブエフェクト(図10(b))はSpiralTVと、残像エフェクト(図10(c))はStreakTVとよく似ている。

EffecTVと本手法の大きな違いは、EffecTVの場合はカメラからの入力画像全体にエフェクトを掛けるのに対し、本手法はARで描画した立方体の領域上だけにエフェクトを掛けることである。EffecTVは画面全体にエフェクトが掛かるので、「魔法の鏡」のよう

に機能し、映し出される映像効果そのものを楽しむインタラクションとなる。一方、本システムの場合、画面内のキューブ領域に対してのみエフェクトが掛かるため、エフェクトそのものを楽しむというよりもむしろ、現実と見比べることで現実との乖離を楽しむインタラクションとなる。

これは、キューブ領域外に描画されるものは現実世界と同じであり、キューブ領域内に描画されるものとは対比して観察できることで、そこに「効果を及ぼす何か」があるように見えるからだと思われる。例えば図9では過去バッファをキューブ領域にのみ重畳するが、これを画面全体に対し過去バッファを重畳し提示するようにしたならば、それは「画面上の時間が遅れている」というよりも、単に「処理が遅延してカメラ入力画像を映し出すのに時間が掛かっている」ように見えるだろう。遅延エフェクトが不思議に見えるのは、現実と同じように時間が流れる中で一部の領域だけが時間が遅れて見えるためであり、これによりその領域だけが現実とずれているように見える効果が発生する。これは、これまであった様々な錯覚表現のように、我々が知覚する世界との奇妙なずれがその面白さを表出させているのだと思われる。

4 オブジェクトの透視表現

2章, 3章で用いた手法を組み合わせることで、任意の領域に対してその領域内だけARオブジェクトが切り替わるような表現ができる。例えば図11の出力画像には、全く同じように見える赤い箱のARオブジェクトに対してキューブ領域を重ねることで、それぞれの箱の中身が違うことが観察できる。

4.1 関連研究

中村らは、眉間付近に付けたフォトリフレクタで眉間の動きを取得し、それによってARの情報量を制御するインタフェイスを開発した[28]。井川らは、今まで固定的だったARマーカのパターンを組み換え可能にすることで一つのマーカで複数のオブジェクトを切り替えて描画することを可能にした[29]。本研究による手法は、これらのようにARで表現する情報を制御する手法のうちの一つであると言える。中村らの手法や井川らの手法では、情報の変化の仕方はシステム開発者に委ねられる。一方本手法では、画像を歪ませるためのソースとして、オブジェクトの形状ではなくオブジェクトに貼り付けられたテキストの色情報を用いる。そのため、オブジェクトの形状と画像の歪ませ方を別々の情報として与えることができる。

4.2 アルゴリズム

アルゴリズムの概要は図11に示す。まずカメラからの入力画像をバッファに記録する(図11(a))。次に、その入力画像を用いて、一つ目と二つ目のマーカ上

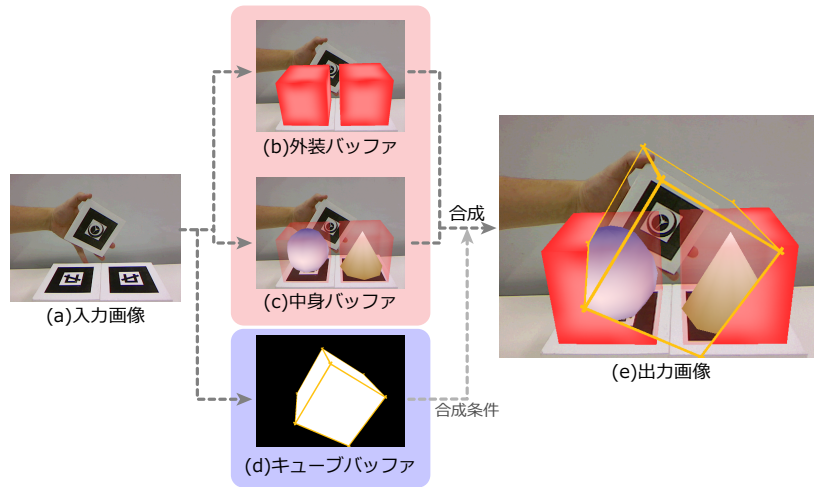


図 11: オブジェクトの透視表現のアルゴリズム概念図

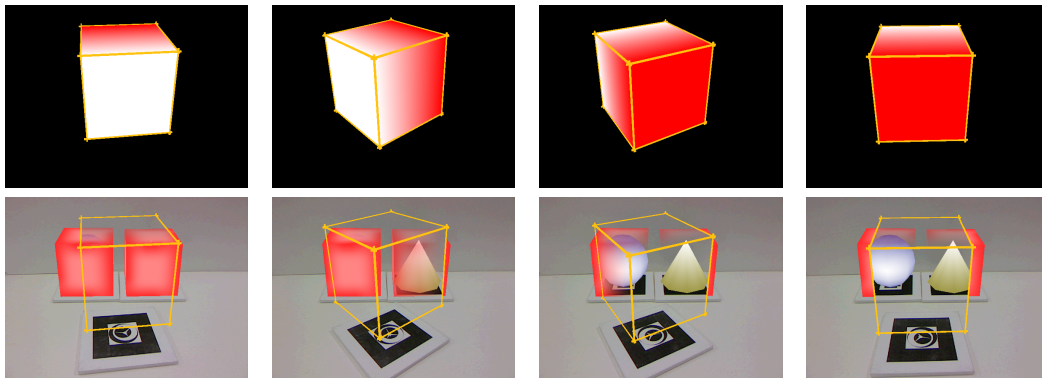


図 12: AR オブジェクトが徐々に切り替わる表現 上段：キューブバッファ 下段：出力画像

に全く同じ赤い箱のオブジェクトを描画し(図 11(b)),これを「外装バッファ」と呼ぶことにする。その次に、入力画像を再度用いて、一つ目のマーカ上には半透明の赤い箱の中に青い球があるようなオブジェクトを、二つ目のマーカ上には半透明の赤い箱の中に黄色い円錐があるようなオブジェクトを描画し(図 11(c)),これを「中身バッファ」と呼ぶことにする。さらにその次に、入力画像を用いて三つ目のマーカ上に、黒(R:0, G:0, B:0)の背景に白(R:255, G:255, B:255)の面、オレンジ色(R:255, G:194, B:14)の辺を持つ立方体を描画する(図 11(d))。これを「キューブバッファ」と呼ぶことにする。出力画像は、「キューブバッファ」のすべての画素を走査し、その画素の色が黒だったならば「外装バッファ」の同座標の画素の色を、白だったならば「中身バッファ」の色を、オレンジ色だったならば「キューブバッファ」の色をそれぞれ描画する。これにより、立方体が重なっている部分だけ箱の中身が見えるような画像が得られる(図 11(e))。

4.3 応用

この手法を応用することで、マーカの回転や大きさ、種類を変化させるといった従来のマーカ式 AR のインタラクションに加え、オブジェクトの任意領域に対して干渉し鑑賞するというインタラクションが可能になる。例えば、家の模型オブジェクトを描画する AR に対してキューブをかざすことで任意の領域だけ屋根が無いかのように見せて間取りを観察したり、植木鉢が描画された AR に対してキューブを重ねるとそこに花が現れる、などといった応用手法が考えられる。

また、この手法をさらに応用し、単にバッファ同士を切り貼りあわせるのではなく、「外装バッファ」と「中身バッファ」をなめらかにブレンドすることで徐々にオブジェクトが切り替わるような表現(図 12(下段))ができる。これには、図 12(上段)のようなグラデーション状のテクスチャを貼った立方体のオブジェクトを用い、この色を画像のブレンドの重み付けに利用することで実装している。

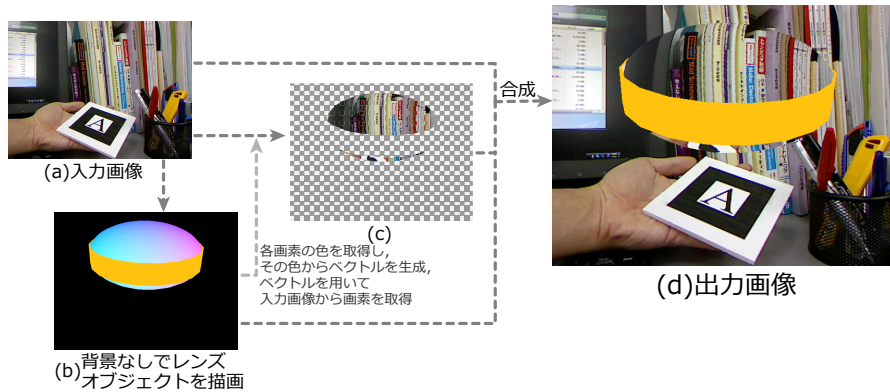


図 13: 動的な屈折表現のアルゴリズム概念図

具体的なアルゴリズムは以下のとおりである．図 12(上段) を見て分かる通り，立方体の面は白から赤へ変化するグラデーションのテクスチャを持っている．より詳しく言えば，この色は白 (R:255, G:255, B:255) から赤 (R:255, G:0, B:0) へと変化する．従って，この色の緑要素 (青要素でも構わない) は 255 から 0 へと線形に遷移する「キューブバッファ」の立方体の面上のある画素 P_i の色を (R: r_i , G: g_i , B: b_i) とし、「外装バッファ」の同座標の画素の色を (R: r'_i , G: g'_i , B: b'_i)，「中身バッファ」の同座標の画素の色を (R: r''_i , G: g''_i , B: b''_i) と置く．出力画像の同座標の画素の色 (R: r'''_i , G: g'''_i , B: b'''_i) を次の式で求める：

$$w = g_i/255$$

$$r'''_i = wr'_i + (1.0 - w)r''_i$$

$$g'''_i = wg'_i + (1.0 - w)g''_i$$

$$b'''_i = wb'_i + (1.0 - w)b''_i$$

「キューブバッファ」上の画素の色が白から赤に近づくにつれて w は 1.0 から 0.0 へ遷移するため，これにより，ブレンドの重みが「外装バッファ」から「中身バッファ」へと移っていくことになる．これは即ち，立方体に貼られたグラデーションテクスチャの色が「外装バッファ」と「中身バッファ」のブレンド率を表す，ということである．今回の実装では線形にグラデーションするテクスチャを立方体状のオブジェクトに貼り付けていたが，テクスチャのグラデーションの仕方を変えたり，テクスチャを貼るオブジェクトの形状を別のものにすることで，バッファのブレンドの仕方を様々なものとする事も可能である．

5 動的な屈折表現

4 節のように，別々のバッファ同士を合成するのではなく，一つのバッファの別の座標から色を持つことで，図 13 の出力画像のように，オブジェクトがあたかもレンズであるかのような画像を生成することができる．

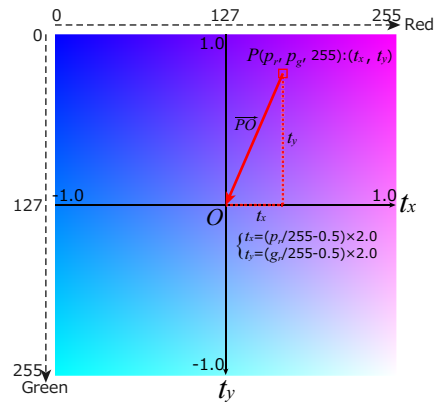


図 14: 色情報から生成されるベクトル

5.1 関連研究

橋本は，AR で描画したオブジェクトのデプスバッファを用いてそのオブジェクト状に画像を歪ませることで，光学迷彩のように見せる手法を [30] で紹介している．また，オブジェクトに対して屈折率を設定し背景を適切に歪ませることで，オブジェクトの材質がガラスであるかのように見せることができる表現 [31][32] もある．これらの手法は本手法よりも歪みの形としては正確であり，例えばオブジェクトを回転させても歪み方が変化したりはしない．一方本手法では，画像を歪ませるためのソースとして，オブジェクトの形状ではなくオブジェクトに貼り付けられたテクスチャの色情報を用いる．そのため，オブジェクトの形状と画像の歪ませ方を別々の情報として与えることができる．



図 15: レンズを真正面から見た場合

5.2 アルゴリズム

このアルゴリズムの概要は図 13 に示す。まず、入力画像 (図 13(a)) からマーカ位置を取得し、それを元に、背景色を黒 (R:0, G:0, B:0) にしてレンズ状のオブジェクトを描画する (図 13(b))。レンズの枠の内側は、色の範囲が (R:0 ~ 255, G:0 ~ 255, B:255) のテクスチャが貼ってあり、レンズの枠の画素の色はオレンジ色 (R:255, G:194, B:14) である。レンズオブジェクトを描画したバッファのすべての画素を走査し、その画素の青要素が 255 である (レンズの内側) ならば、以下の計算を行う。

その画素を P とし、その x 座標と y 座標をそれぞれ x, y とする。画素 P の赤要素と緑要素をそれぞれ p_r, p_g とし、それらを独立変数とする関数を以下のように定義する：

$$t_x = 2(p_r/255 - 0.5)$$

$$t_y = 2(p_g/255 - 0.5)$$

p_r, p_g は 0 以上 255 以下のため、 t_x, t_y は共に -1.0 以上 1.0 以下の範囲の値を取る。 t_x, t_y を軸とする二次元平面を考えると、その原点 O は (R:255/2, G:255/2, B:255) にあり、これはレンズの内側の中心の色でもある。 \vec{PO} は当然ながら、原点を離れるほど大きくなる原点方向を向いたベクトルとなる (図 14)。

Q を $P - s\vec{PO}$ と定義すると、 Q の座標は $(x + t_x s, y + t_y s)$ である。ここで s はレンズの擬似的な歪みの大きさを表す比例定数である。例えば P の色が (R:153, G:51, B:255) であったならば、 Q の座標は $(x + 0.2s, y - 0.6s)$ となる。

そして、レンズの内側に重畳するバッファの P に Q を代入する (図 13(c))。 \vec{QP} は $s\vec{PO}$ と等しいためレンズの中心向きのベクトルとなる。従ってレンズの内側に重畳するバッファは入力画像を s に応じて縮小したような画像となるため、結果として凹レンズを通して見たかのような画像が得られる (図 13(d))。この凹レンズを通してかのように縮小されて見える効果は、真正面からレンズを見ることでより分かりやすく感じられる (図 15 左)。

図 15 は、解像度を幅 640 画素、高さ 480 画素、比例定数 s は 50.0 として描画している。図 15 上の正方形の中心の画素 P の色は (R:153, G:51, B:255) であり、この画素における Q の座標は $(x + 0.2 \times 50.0, y - 0.6 \times 50.0) = (x+10, y-30)$ となる。結果、この画素 P



図 16: 様々なテクスチャを貼ったレンズ

は入力画像より右に 10、上に 30 の画素を出力するため、図 15 に示す矢印の方向に歪められる。なお、図 15 上の矢印とその先にある正方形の枠はそれぞれ、とある画素における \vec{QP} と P であり、本来の出力画像には含まれない。

マーカを回転させると、それに伴ってオブジェクトも回転する。これによって、 \vec{QP} の向きも回転する。従って、例えばオブジェクトを 90 度回転させると、レンズ内領域は拡大も縮小もしないがベクトルの向きにひねられたかのような出力画像となり (図 15 中央)、180 度回転させると \vec{QP} はレンズの外側を向くため、凸レンズで拡大されたかのような出力画像となる (図 15 右)。

5.3 応用

本手法は、実際のレンズによる光の屈折を正しくシミュレートしているわけではない。また、常に凹レンズ (凸レンズ) のように見えるオブジェクトを作るこ

ともできない。だが一方で、回転させるだけで凹レンズのように凸レンズのようにも見える、現実ではありえないようなレンズを作り出すことができるという特徴がある。

このアルゴリズムでは、 \vec{QP} はオブジェクトが描画された画像バッファの色から定められる。そのため、レンズに貼るテクスチャ自体に効果を加えておくことで、単にレンズ状ではない歪み方をするようなオブジェクトを作ることも可能である (図 16)。

これらのように、この手法は、あるオブジェクトに対して行う処理を、それに貼りつけたテクスチャを用いてマッピングすることができる。本論文で挙げた作例では、入力画像をレンズ状に歪めることで光学的な屈折を起こしているように見せたが、例えばこの手法を 3 章の手法に応用し、レンズに貼りつけたテクスチャの色によって画素に代入する過去フレームを定めれば、レンズ状に時間を歪めるような表現ができる。また、カメラを固定し、背景バッファを予め取得しておくことで、マーカ部分を背景に置換し隠すことも可能である。

6 おわりに

本論文では、複数の画像バッファをリアルタイムに合成することで錯覚表現を作り出す手法について述べ、そのアルゴリズムとそれを用いた作例を示した。

本論文に用いた手法は「バッファの色情報を用いてその情報を書き換える」という極めて平面的な処理しか行っていない。にもかかわらず、不可能立体の AR 表現ではオブジェクトの奥行きが入れ替わったかのように見え、時空間を歪める錯覚表現におけるキューブ領域は画面上の平面的な領域でありながら、あたかも立体的な領域であるかのように観察される。

本手法は非常に簡素でありながら、様々な処理に適用することができる。本論文に示した作例以外の利用法も多数存在するだろうと思われる。本手法が、人が思い描いたイメージを表現するための方法として用いられ、そして人のイメージを喚起させて新たな表現が生まれる可能性に期待したい。

参考文献

- [1] 神原誠之：拡張現実感 (AR) : 1 . 基礎 1 : 拡張現実感 (Augmented Reality : AR) 概論, 情報処理, Vol. 51, No. 4, pp. 367–372 (2010).
- [2] 杉原厚吉：だまし絵のトリック 不可能立体を可能にする, 化学同人 (2010).
- [3] 吉川祐輔, 宮下芳明：不可能立体の拡張現実感表現, 第 10 回 NICOGRAPH 春季大会 CD-ROM 予稿集, pp. 162–163 (2011).
- [4] 吉川祐輔, 宮下芳明：複数の画像バッファを組み合わせることによる拡張現実感上での錯覚表現, インタラクシオン 2012 論文集, pp. 777–782 (2012).
- [5] : 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 (2’28’).
<http://www.youtube.com/watch?v=-H9PUdyHV-I>.
- [6] 今井省吾：錯視図形—見え方の心理学—, サイエンス社 (1984).
- [7] 杉原厚吉：立体イリュージョンの数理, 共立出版 (2006).
- [8] Khoh, C. W. and Kovese, P.: Animating Impossible Objects.
<http://www.csse.uwa.edu.au/~pk/Impossible/impossible.html>.
- [9] Khoh, C. W. and Kovese, P.: Rotating the Impossible Rectangle, *Leonardo*, Vol. 34, No. 3, pp. 197–198 (2001).
- [10] Owada, S. and Fujiki, J.: DynaFusion: A Modeling System for Interactive Impossible Objects, *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pp. 65–68 (2008).
- [11] 藤木 淳, 牛尼剛聡, 富松 潔: インタラクティブだまし絵表現の提案と実装, 情報処理学会研究報告. グラフィクスと CAD 研究会報告, Vol. 2006, No. 119, pp. 31–36 (2006).
- [12] ソニー・コンピュータエンタテインメント: 無限回廊.
<http://www.jp.playstation.com/scej/title/mugen/mugen/>.
- [13] Savransky, G., Dimerman, D. and Gotsman, C.: Modeling and Rendering Escher-Like Impossible Scenes, *Computer Graphics Forum*, Vol. 18, pp. 173–179 (1999).
- [14] 篠原祐樹, 宮下芳明: 不可能立体のレイトラッキング, インタラクシオン 2009 論文集, Vol. 2009, No. 4, pp. 143–144 (2009).
- [15] 篠原祐樹, 宮下芳明: 不可能立体の写実的表現手法の提案, 情報処理学会研究報告. HCI, ヒューマンコンピュータインタラクシオン研究会報告, Vol. 2009, No. 28, pp. 95–102 (2009).
- [16] 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*, Vol. 29, No. 2, pp. 13:1–13:15 (2010).
- [17] McCann, J. and Pollard, N.: Local layering, *ACM SIGGRAPH 2009 papers*, SIGGRAPH ’09, ACM, pp. 84:1–84:7 (2009).
- [18] 禹 承賢, 青木孝文, 三武裕玄, 橋本直己, 佐藤 誠: 鏡の錯覚に着目したインタラクティブエンタテインメントの実現: Heaven’s Mirror, インタラクシオン 2008 論文集, Vol. 2008, No. 4, pp. 147–148 (2008).
- [19] 筧 康明, 苗村 健: through the looking glass, 芸術科学会論文誌, Vol. 3, No. 3, pp. 185–188 (2004).

- [20] Igarashi, T. and Mitani, J.: Apparent layer operations for the manipulation of deformable objects, *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, New York, NY, USA, ACM, pp. 110:1–110:7 (2010).
- [21] 高嶋章雄, 山本恭裕, 中小路久美代: 探索的データ分析のための時間的な概観と詳細の表現およびインタラクションに関する研究, *情報処理学会論文誌*, Vol. 44, No. 11, pp. 2767–2777 (2003).
- [22] Cassinelli, A. and Ishikawa, M.: Khronos projector, *ACM SIGGRAPH 2005 Emerging technologies*, SIGGRAPH '05, ACM (2005).
- [23] Cassinelli, A.: KHRONOS PROJECTOR.
<http://www.k2.t.u-tokyo.ac.jp/members/alvaro/Khronos/>.
- [24] Matsuzaki, K., Iwai, D. and Sato, K.: Snail light projector: interaction with virtual projection light in hyper-slow propagation speed, *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*, ACE '10, ACM, pp. 114–115 (2010).
- [25] 藤田真吾, 吉野 孝: ドアコム: タンジブルインタフェースを用いたビデオチャットシステム, *インタラクション 2011 論文集*, Vol. 2011, No. 3, pp. 757–760 (2011).
- [26] 福地健太郎, Tannenbaum, E.: EffecTV: メガデモ技術のリアルタイムビデオイフェクトへの応用, *エンタテインメントコンピューティング 2003 論文集*, Vol. 2003, No. 1, pp. 94–99 (2003).
- [27] EffecTV crew: EffecTV.
<http://effectv.sourceforge.net/index.ja.html>.
- [28] 中村裕美, 宮下芳明: F.A.R. Vision: 拡張現実感の情報量を制御する眉間フェーダー, *インタラクション 2010 論文集*, Vol. 2010, No. 4, pp. 177–180 (2010).
- [29] 井川洋平, 宮下芳明: 組み換え可能な AR マーカの提案, *エンタテインメントコンピューティング 2011 予稿集*, pp. 218–221 (2011).
- [30] 橋本 直: 3D キャラクターが現実世界に誕生! AR-ToolKit 拡張現実感プログラミング入門, 株式会社アスキーメディアワークス (2008).
- [31] Pessoa, S., Moura, G., Lima, J., Teichrieb, V. and Kelner, J.: Photorealistic rendering for Augmented Reality: A global illumination and BRDF solution, *Virtual Reality Conference (VR), 2010 IEEE*, pp. 3–10 (2010).
- [32] Aittala, M.: Inverse lighting and photorealistic rendering for augmented reality, *The Visual Computer*, Vol. 26, pp. 669–678 (2010).

吉川 祐輔



2010 年明治大学理工学部情報科学科卒業, 2012 年明治大学大学院理工学研究科新領域創造専攻デジタルコンテンツ系博士前期課程卒業. 2012 年修士 (工学) 取得.

宮下 芳明



千葉大学工学部卒業 (画像工学), 富山大学大学院で音楽教育 (作曲) を専攻, 北陸先端科学技術大学院大学にて博士号 (知識科学) 取得, 優秀修了者賞. 2007 年度より明治大学理工学部に着任. 2009 年度より准教授, 2013 年度より総合数理学部先端メディアサイエンス学科に移籍, 現在に至る. 専門はヒューマンコンピュータインタラクション (HCI). 情報処理学会インタラクション 2012 プログラム委員長など.