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

藤木 淳† 大和田 茂‡

†九州大学 〒812-8581 福岡市東区箱崎 6 丁目 10 番 1 号
‡Sony CSL 〒141-0022 東京都品川区東五反田 3-14-13 高輪ミューズビル
E-mail: †jun_Fujiki@hotmail.com, ‡owd@csl.sony.co.jp

あらまし 一つのシーンでも，視点によって印象は変わる．theRelativity はこの考えを深くつきつめて，三人称視点で見た場合に連想できる立体構造を一人称視点から見たシーンに反映し，一人称視点から見た形状に対する振る舞いとの辻褄が合うように三人称視点のシーンにシームレスに反映するインタラクティブアート作品である．本稿ではこのコンセプトに基づく４つの表現の紹介とコンピュータで実行するための技法について述べる．

キーワード 認知，相対性，だまし絵，エッシャー

theRelativity
－An Technique of Illusionary Picture Expression Reflected A Difference Between First Person View and Third Person View－

Jun FUJIKI‡ and Shigeru OWADA‡

†Kyushu University 6-10-1, Hakozaki, Higashi-ku, Fukuoka, 812-8581 JAPAN
‡Sony CSL 3-14-13, Takanawa Muse Bldg, Higashigotanda, Shinagawa-ku, Tokyo, 141-0022 JAPAN
E-mail: †jun_Fujiki@hotmail.com, ‡owd@csl.sony.co.jp

Abstract An impression changes for one scene depending on the aspect. We deeply stand for this idea. theRelativity is an interactive art work that seamlessly reflected a structure associated in the scene of the third person view into the scene of the first person view. And also it consistently reflected the behavior in the scene of the first person view into the scene of the third person view. In this paper, we describe the computer programming technique and four expressions based on this concept.

Keyword Cognitive, Relativity, Illusionary Picture, Escher

1. はじめに

アインシュタインは相対性理論として絶対的でない時間の理論を打ち出したが，我々は theRelativity においてユーザが見るシーンとキャラクタから見たシーンにギャップを生じさせることでユーザに個人によって同じ距離も異なることを認識させる機会とする．

我々は前研究において M.C.Escher[1]の作品に見られるような，だまし絵をモチーフとしたインタラクティブ作品を提案した[2]．だまし絵表現は人間が空間認知を誤ることにより起こるものであるため，人間の直感からするとおかしな結果が本当は正しい場合もあるかもしれない．

theRelativity において，我々が提案する表現から，直感的には自分が絶対的だと感じている我々を取り巻く様々な要素（時間，距離，速度など）の関係を見つめ直す機会となることを期待する．

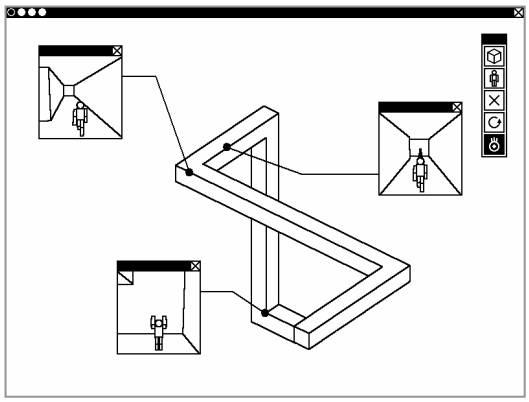

図 1. theRelativity の画面

2. システム

theRelativity の画面の例を図 1 に示す．ユーザは三人称視点でカンバス（以下、ユーザビュー）にマウス操作で複数のブロックを生成して立体形状を作成した

り，作成した立体形状内にキャラクタを配置したり，視点を回転移動することができる．配置したキャラクタは黒丸で表示され，ユーザは目アイコンで黒丸をクリックして，キャラクタの視点から見た光景を別ウィンドウ（以下、キャラクタビュー）で表示することができる．キャラクタは自発的に壁に沿って移動したりジャンプしたりする．各キャラクタは，ユーザビューにおいて自然に感じられる接続性を保った，キャラクタごとに局所的に定義された世界を歩き回る．この世界が３次元的に大局的に見て一貫したものである必要はない．キャラクタの移動に関しても，ユーザビューで見た自然さが常に維持されるようになされる.また，キャラクタが３次元空間では繋がっていないがユーザビューで見たときに自然に感じられるブロック間を移動後，ユーザビューからの見た目を保ったまま，キャラクタが直前にいたブロックを現在いるブロックの方へ３次元移動したり，直前にいたブロックと現在いるブロックが繋がるように変形したりすることができる.ユーザは視点を回転することでそのことを確認することができるが，キャラクタビューにおいてはその変化を確認できない．キャラクタから見た世界をユーザから見た世界にシームレスに反映することで，ユーザの視点から見た世界とキャラクタから見た世界が交換される瞬間である．この一見正しいように見えて正しくない見え方が内包する不思議な不自然さ，視点による非一貫性がこのコンテンツの最大のメッセージである.本システムはこのようなコンセプトを持つ４つのモードがある．モードの変更は画面左上部のスイッチボタンで行う．以下でそれぞれのモードについて述べる．

2.1. オリジナルモード

キャラクタは３次元空間では繋がっていないがユーザビューでは繋がっているように見えるブロック間を移動でき，移動後に視点回転するとやはり形状は繋がっていないことをユーザは確認できる（図 2）．キャラクタはユーザビューにおいて繋がっているように見えるときのみ移動できる．

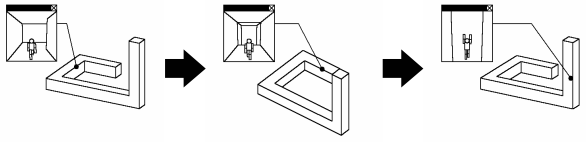

図 2. オリジナルモード

2.2. 形状移動モード

キャラクタが３次元空間では繋がっていないがユーザビューでは繋がっているように見えるブロック間を移動後，システムはキャラクタが直前にいたブロックをユーザビューでの見た目を変化しないようにキャラクタが現在いるブロックに接するように移動する（図 3）．ユーザは視点を回転してブロックの移動を確認できる．

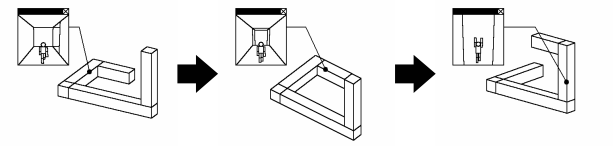

図 3. 形状移動モード

2.3. 形状変形モード

キャラクタが３次元空間では繋がっていないがユーザビューでは繋がっているように見えるブロック間を移動後，システムはキャラクタが直前にいたブロックをユーザビューでの見た目を保ったままキャラクタが現在いるブロックに接するように変形する（図 4）．ユーザは視点を回転してブロックの変形を確認できる．キャラクタが変形したブロック内を移動するとき，キャラクタビューでは変形前のブロックが表示され，キャラクタはこのキャラクタビューに見える形状に対して振舞う．ユーザビューには変形前に位置対応する変形後の位置に黒丸が表示される．

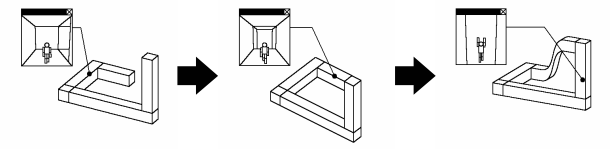

図 4. 形状変形モード

2.4. 不可能物体モード

キャラクタが３次元では繋がっていないがユーザビューでは繋がっているように見えるブロック間を移動後，視点を回転しても常に見た目の接続関係を維持して変形するようになる（図 5）．ユーザはそのように変形した形状を不可能物体と認識するかもしれない．変形モードと同様に，ユーザビューでのキャラクタの位置は変形後に位置対応する．

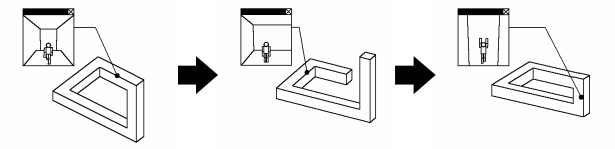

図 5. 不可能物体モード

```
Function Display( ViewPosition, SearchPosition, Nest )
{
        if( Nest ==0 )
                return;

        ViewPosition にブロックを表示;

        if( SearchPosition から１ブロック前方に隣接ブロックがある )
                Display( ViewPosition を前方に進める，前方隣接ブロックの３次元位置, Nest-1 );
        if( SearchPosition から１ブロック左方にブロックがある )
                Display( ViewPosition を左方に進める，左方隣接ブロックの３次元位置, Nest-1 );
        if( SearchPosition から１ブロック右方にブロックがある )
                Display( ViewPosition を右方に進める，右方隣接ブロックの３次元位置, Nest-1 );
        if( SearchPosition から１ブロック後方にブロックがある )
                Display( ViewPosition を後方に進める，後方隣接ブロックの３次元位置, Nest-1 );
        if( SearchPosition から１ブロック上方にブロックがある )
                Display( ViewPosition を上方に進める，上方隣接ブロックの３次元位置, Nest-1 );
        if( SearchPosition から１ブロック下方にブロックがある )
                Display( ViewPosition を下方に進める，下方隣接ブロックの３次元位置, Nest-1 );
}
```

図 7 表示のための擬似プログラミングコード

3. 実装

本表現を実現するために重要となるキャラクタの移動とキャラクタビューにおける表示の実装方法について述べる．本稿では，３次元座標上で隣接しているブロックを実在隣接ブロック，ユーザビューにおいて隣り合っているように見えるブロックを主観隣接ブロック，実在隣接ブロックと主観隣接ブロックをまとめて隣接ブロックと呼ぶことにする．

3.1. オリジナルモード

キャラクタは左手法[3]により左折を優先しながら隣接ブロックの中央位置を目指して移動する．主観隣接ブロックは，本来３次元座標上で隣接する場合のブロックのスクリーン位置と近い位置にあるスクリーン位置を持つブロックとする．中央位置に到達したブロックが主観隣接ブロックだった場合，キャラクタの３次元位置を主観隣接ブロックの３次元位置に移動する（図 6）．

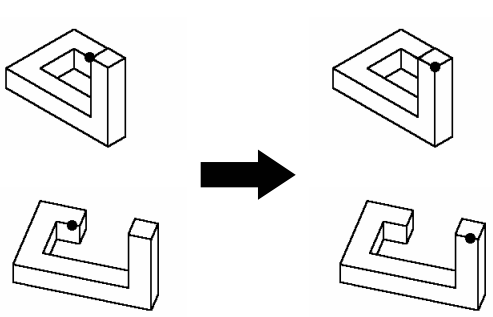

図 6. キャラクタの移動（上：ユーザビュー、
下：斜め左上方向から見た視点）

キャラクタビューにおける表示は，まず，現在の探索位置（SearchPosition）から前後左右上下方向の隣接ブロックを探索する．見つかった場合はキャラクタビューの表示すべき位置（ViewPosition）にブロックを表示し，ViewPosition と SearchPosition を更新する．見つかった隣接ブロックが主観隣接ブロックの場合は SearchPosition を主観隣接ブロックの３次元位置にする．以下、再帰的にこの処理を繰り返す．最初の探索位置はキャラクタの３次元位置である．上記の処理をＣ言語に似た擬似プログラミングコードを図 7 に示す．

3.2. 形状移動モード

形状移動モードでは，直線状に隣接するブロック群を一つのグループとする．オリジナルモードに加え，キャラクタが移動した隣接ブロックが主観隣接ブロックだった場合に，キャラクタが移動前にいたグループに属するブロック群をキャラクタの移動距離だけ移動させる（図 7）．

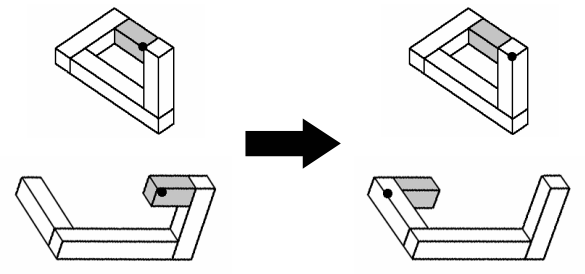

図 7. キャラクタと共に移動する形状
（上：ユーザビュー、下：横方向から見た視点、
灰色ブロック：移動対象ブロック）

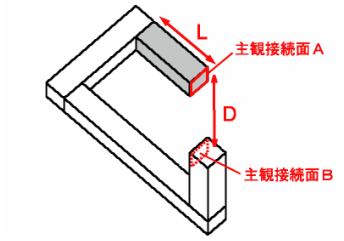

図 8. 斜め上方向から見た立体形状

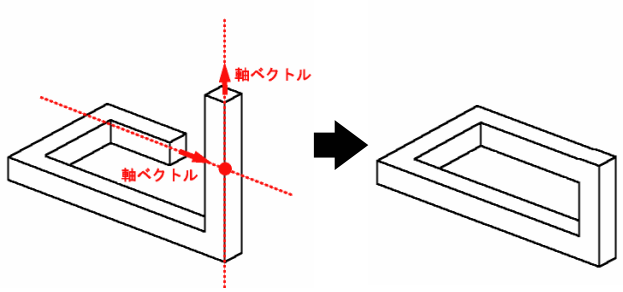

図 9. 交点位置まで伸縮するグループ直方体

3.3. 形状変形モード

形状変形モードにおいても，直線状に隣接するブロック群を一つのグループとする．オリジナルモードに加え，キャラクタが移動した隣接ブロックが主観隣接ブロックだった場合に，キャラクタが移動前にいたグループに属するブロック群を削除し，ユーザビューでの見た目が変化しないように移動後のブロックの３次元位置に滑らか繋がる形状を生成する．

ここで，ユーザビューでは接続しているように見える２つのブロックの互いに向き合う面を主観接続面（A,B）と呼ぶことにする（図 8）．接続面間距離を D，変形対象のグループの長さを L とし，V と P1, P2, P3, P４を下記のように定める．

V = D×(0.25-0.25/(L+2))
P1：主観接続面 A
P2：主観接続面 A＋主観接続面 A の法線×V
P3：主観接続面 B＋主観接続面 B の法線×V
P4：主観接続面 B

P1，P2，P3，P４をパラメタに持つベジェ曲線から求める３次元形状を算出する．

3.4. 不可能物体モード

オリジナルモードに加え，キャラクタが移動した隣接ブロックが主観隣接ブロックだった場合に，キャラクタが移動前にいたグループに属するブロック群と移動後のグループに属するブロック群を削除し，それぞれ新たなブロック（以下、不可能ブロック）を生成する．このとき，各グループ内の先頭ブロックの中央位置から後方ブロックの中央位置に向かう方向ベクトルを軸ベクトルとして保持しておく．視点に依存して不可能ブロックがユーザビューでの見た目の接続を維持するようにブロックを変形させる．具体的な手順を以下に述べる．

１．ユーザビューにおいて，それぞれ不可能ブロックの中央を通り軸ベクトルに平行な直線を引き交点を算出する（図 9 左）．

２．３次元座標において，1 で求めた交点を通り視線方向ベクトルと平行なベクトルと各軸ベクトルとの交点を求める．

３．2 で求めた各交点位置に接するように不可能ブロックを伸縮する（図 9 右）．

4. まとめ

本稿では，theRelativity におけるユーザに個人によって同じ距離も異なることを認識させるコンセプトに基づく４つ表現と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技法について述べた．これらの提案表現，提案技法が我々を取り巻く様々な要素の関係を見つめ直す機会となることを期待する．我々は今後も人間と我々を取り巻く様々な要素の関係を見つめなおす表現を模索する．

文　献

[1] Bruno Ernst, 坂根厳夫: エッシャーの宇宙，朝日新聞社出版局(1983)

[2] 藤木 淳，牛尼 剛聡，富松 潔：“２次元動画像に対する３次元解釈の視知覚特性を利用したインタラクショブだまし絵", 情報処理学会論文誌 Vol.48，No.12，pp.3765-3771（2007 年）

[3] 芝浦工業大学ロボティクス研究室: 左手法 http://www.robotics.ee.shibaura-it.ac.jp/manual/chap4/L HANDexp.htm