

コンピュータ工学特別研究報告書

題目

実写立体 360°画像への 3次元CG
オブジェクトの自然な合成に関する研究

学生証番号 444215

氏名 江原貴大

提出日 平成30年1月24日

指導教員 蚊野 浩

京都産業大学
コンピュータ工学部

要約

現実世界の情報と仮想世界の情報を複合させる技術が娯楽だけでなく、医療や教育など様々な分野で活用されるようになってきている。この技術の一部に立体 360°画像というものがある。本研究ではこの立体 360°画像に違和感なく 3次元 CG キャラクタを重畳して合成表示するためにどういった要素が有効であるかを検証した。

両眼立体視には両眼視差が重要である。両眼視差とは人の右目と左目が見ている像のズレで、近距離の立体視では最も重要な要素となっている。これに対し、立体360°画像は、両眼視差はあるが画像を表示しているだけなので、明示的な3次元情報を持っていない。このため、合成する3次元CGキャラクタの位置・両眼視差と立体360°画像の位置・視差の関係に違和感が生じてしまう場合がある。この両眼視差・位置関係の二点において違和感のない合成手法を検証した。

位置関係の調整には、距離画像を使用した。右目側と左目側で使用している画像からステレオマッチングを行い、距離マップを作成する。この距離画像から、3次元CGキャラクタが違和感の生じる位置に向かわないように移動を制限した。

両眼視差の調整については、立体360°画像の視線方向が回転する場合に、3次元CGキャラクタに与える両眼視差を調整する必要があることがわかった。これを解決するために、カメラの回転角から、3次元CGキャラクタの位置を調整するアルゴリズムを実装した。

2点を実装した結果、ある程度の違和感を取り除くことに成功した。しかし、陰影など他にも違和感を発生させている要素があり完全に違和感を取り除く結果とはならなかった。

Google のストリートビューのように視点がある程度自由に移動できるような立体 360°画像にこのような合成が可能になれば、建造物のある敷地に建てようとした場合や、部屋内に新しい家具を設置しようとした場合などのシミュレーションなど実用的な部分での、より現実的でわかり易い想定が可能になるだろうと考えられる。

目次

| | |
|--------------------------------|--------|
| 1 章 序論 | ．．． 1 |
| 2 章 予備知識 | ．．． 2 |
| 2.1 仮想と現実と本研究の関係 | ．．． 2 |
| 2.2 使用機材 | ．．． 3 |
| 2.3 立体 360° 画像について | ．．． 4 |
| 2.4 距離画像について | ．．． 4 |
| 3 章 提案手法について | ．．． 5 |
| 3.1 提案手法の概要 | ．．． 5 |
| 3.2 CG オブジェクトの動作の実装 | ．．． 6 |
| 3.3 距離画像の生成と CG オブジェクトの行動範囲の制限 | ．．． 7 |
| 3.4 CG オブジェクトの両眼視差の調整 | ．．． 8 |
| 4 章 研究結果と考察 | ．．． 12 |
| 5 章 結論 | ．．． 13 |
| 参考文献 | ．．． 14 |
| 謝辞 | ．．． 14 |
| 付録 | ．．． 15 |

1 章 序論

コンピュータ内にモデル化した空間を、人間が体験できるようにする技術を仮想現実感 (VR, Virtual Reality) とよぶ。仮想現実感の主要な要素の一つに視覚情報があり、そこで利用される画像や映像を VR 画像・VR 映像とよぶ。代表的な VR 画像に立体画像や 360° 画像がある。

狭義の VR はコンピュータ内に構築する仮想空間だけを扱う。一方、VR に現実空間の情報を扱うしくみを組み込んだものも存在する。それらは、拡張現実 (AR, Augmented Reality), 複合現実 (MR, Mixed Reality), 仮想化現実などとよばれる。

「Pokemon GO」のように、現実空間と仮想空間を合成して AR を体験できる製品がリリースされている。こういった現実と仮想を組み合わせる技術は娯楽分野だけでなく、医療や製造業、教育分野でも活用されるようになってきている。また、今後は「Microsoft HoloLens」(図 1) などのように、現実空間と仮想空間をリアルタイムに共存させる MR の概念を実現できるシステムにも期待が寄せられている (図 1)。



図 1 Microsoft HoloLens Development Edition

仮想化現実とは、実写の画像・映像からそれらを表現する 3 次元 CG データを生成し、任意視点の映像表現を可能にするなどの技術である [1]。仮想化現実の一部として立体 360° 画像というものがある。これは、360° 画像という 360° 全周囲を画像に収めたものの発展形でもあり、右目 360° 画像と左目 360° 画像を用意することで両眼立体視を可能にしたものである。実写の立体 360° 画像や立体 360° 映像は現実世界を仮想的に体験することを可能にする

本研究では、この立体 360° 画像に違和感なく 3 次元 CG キャラクタを合成表示するためにどのような要素が有効であるかを検討し実証する。

2章 予備知識

2.1 仮想と現実と本研究の関係

本研究は実写画像とCGを合成するものである。この仮想と現実の合成に関する概念として、AR (Augmented Reality) という拡張現実と呼ばれているものや、MR (Mixed Reality) という複合現実と呼ばれているものがある。本研究とこういった現実と仮想を複合させた概念について簡単な説明を行う。

ARとは現実のシーンにCGによって描かれた仮想物体を合成表示する技術のことである。MRとは仮想環境と現実の物理環境を融合する概念のことである。その他に、拡張VRという仮想の空間に現実の情報を合成するという概念や、仮想化現実など仮想と現実の合成の種類によって様々な概念が存在する。これらをすべて単純に表示するために、数学記号の「+」「→」「U」を用いて説明することとする。

まず、ここでのそれぞれの記号の意味を定義する。3種の記号は、

「A+B」…AにBを合成し一体化させる

「A→B」…AをBに変換する

「AUB」…AとBが共存している

とする。注意事項として概念的にA+BとB+Aは意味が違うので完全に同じとは言えないということを記述しておく。この定義をもとに先ほど挙げた概念を表すと、仮想と現実という2種のパラメータが存在するとして、

拡張現実 = 現実 + 仮想

拡張VR = 仮想 + 現実

仮想化現実 = 現実 → 仮想

複合現実 = 現実 U 仮想

という概念であると概ね説明できる。複合現実は、現実と仮想の混合体とされており、拡張現実と拡張VRはこの複合現実の概念に含まれていると言われている[引用]。

本研究をこれら記号の組み合わせで表す場合、

本研究 = (仮想 → 現実) + 仮想

というものになる。拡張VRにさらにCGを合成したものと言える。拡張現実とも拡張VRとも厳密には言い難いが、現実と仮想を合成させていることは確かなので、本研究は複合現実の一部と位置付けられる。

2.2 使用機材

本研究では立体360°画像を表示するためのデバイスとして、HMD (ヘッドマウントディスプレイ) であるOculus Rift (図2)を使用する。Oculus riftは2016年3月から

発売されている HMD である。VR に特化した HMD で、右目用と左目用の 2 枚の有機 EL パネルを有し、それぞれに画像・映像を表示できる。



図 2 Oculus Rift

また、本製品に同梱されている Xbox One Controller (図 3) を本実験の入力デバイスとして使用する。



図 3 Xbox One Controller

2.3 立体 360°画像について

立体 360°画像とは、立体感の感じられる 360°画像のことである。360°画像とはカメラを中心に全方向を撮影し、一枚の画像に収めたものである。現在は、「RICOH THETA S」などの 360°を撮影できるカメラが販売されているので、比較的容易に 360°画像を作成

することができる。しかし、撮影した一枚の 360°画像をそのまま HMD（ヘッドマウントディスプレイ）で右目と左目に表示しただけでは立体感を感じることができない。人は右目と左目が約 6cm 離れている。このズレにより見えている景色にも左右の目でわずかなズレが生じる。このズレを両眼視差と呼ぶ。人の奥行き知覚は、この両眼視差が重要な要因の一つである。つまり、360°画像で奥行きを感じるためには、両眼視差がある 360°画像のペアを用意し、右目と左目に表示させることが必要になる。この両眼視差のある 360°画像を立体 360°画像（または視差画像）という（図 4）。

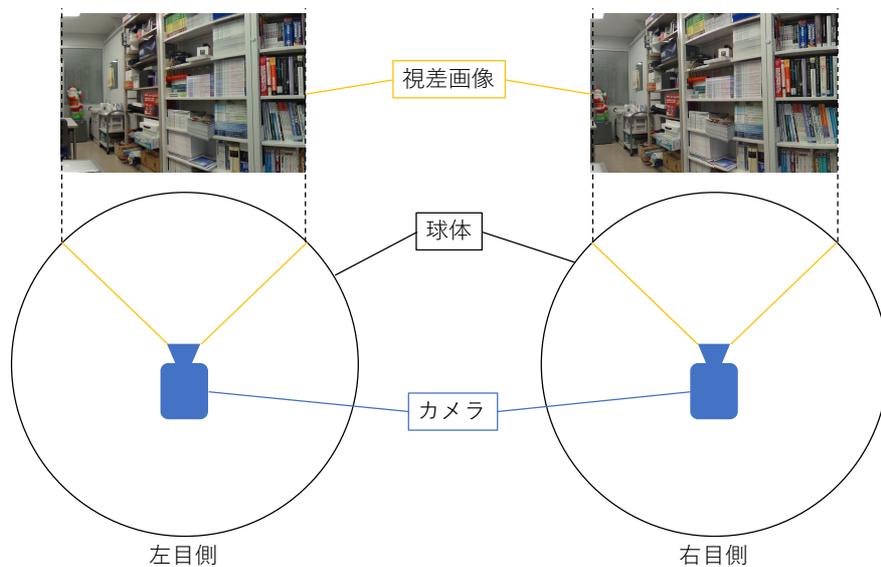


図 4 立体 360°画像

2.4 距離画像について

距離画像とは視差マップとも呼ばれ、例えば、視差のある画像 2 枚からステレオマッチングによって生成することができる。視差は遠くにあるものほど小さく、近くにあるものほど大きくなる。ステレオマッチングはこの視差の性質を利用して、2 枚の画像からテンプレートマッチングを行いマッチングした箇所の視差の大きさを画素値とする方法である。距離画像を濃淡画像として表示すると、遠くにあるものほど暗く表示され、近いものほど明るく表示される。

3章 提案手法について

3.1 提案手法の概要

コンピュータが扱う普通の立体シーン（3次元CGデータ）と比べて、立体360°画像は明示的な3次元情報を含まない。そのため、2枚の360°画像の同じ位置にオブジェクトの画像を重ね書きするような単純な方法では、次のような課題・不具合が生じると推察される。

- (1) 3次元オブジェクトを奥行方向に移動させた場合、360°画像を貼り付けた球面の壁をすり抜ける。（球面よりも遠くに行くと消える）
- (2) 合成した3次元オブジェクトの視差がゼロであれば、両眼視差から判断される距離は無限遠になる。一方、3次元オブジェクトの背面付近の画像は、その距離に応じた視差が存在する。従って両眼視差的には、近い距離にある被写体の前に遠い距離にあるオブジェクトが存在する、という矛盾した状態になる。
- (3) 3次元オブジェクトがシーンに落とす影を生成することが難しい。
- (4) シーン中で手前に存在する物体の裏側に移動することができない。

など。本研究ではこれらの課題の中で(1)と(2)を解決する方法を提案する。

開発環境はゲーム開発プラットフォームのUnityを使用する。Unityのシーン上に右目用・左目用のカメラを配置し、それを囲むように球体オブジェクトを配置する。その球体オブジェクトそれぞれに視差のある360°画像をテクスチャマッピングする。テクスチャマッピングを行った状態のUnityのシーンビューを図5に示す。使用した画像は蚊野教授室で撮影し合成した立体360°画像である。

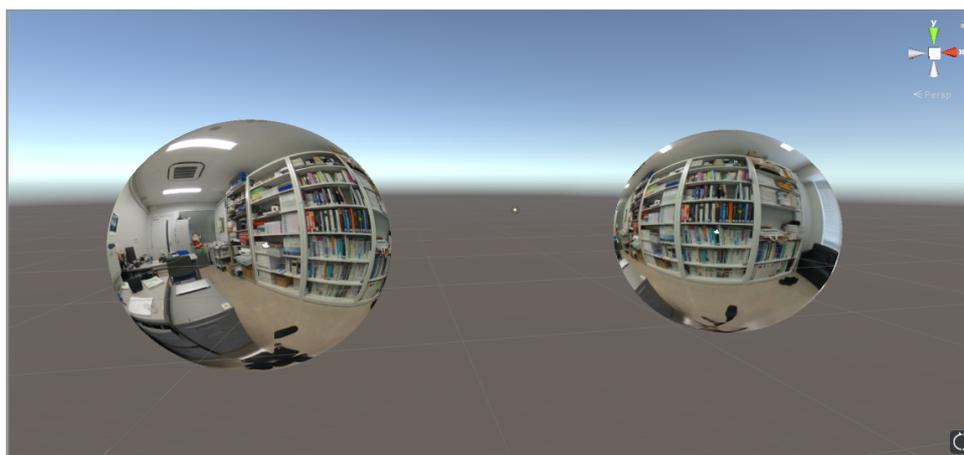


図5 Unityのシーンビュー

この環境にCGオブジェクトの合成を行う。合成するCGオブジェクトはUnity Technologies Japanが開発したオープンソース系アイドルのユニティちゃんのSD版を使用した（図6）。ユニティちゃんの画像も右目用と左目用を用意し、それぞれ配置す

る。この時、右目側と左目側で両眼視差のズレを考慮して配置しなければならない。ユニティちゃんはテクスチャマッピングされている画像の性質上、撮影した画像に写されている物よりも手前にいなければならない。また、視差についても矛盾のないように配置する必要がある。ある物体より、前後関係で見るとユニティちゃんが手前にいる状態でも、視差が逆の前後関係になっていると違和感を感じてしまうからである。



図6 SDユニティちゃん

3.2 CG オブジェクトの動作の実装

配置したユニティちゃんに対して動作を実装する。2.2で紹介したコントローラーを使い、ユニティちゃんが空間の中を自由に動き回れる操作を実装する。後の節で実装する行動範囲の制限の理由から、前後左右だけでなく上下についても移動可能とし、Unityで重力の影響を受けないように設定する。動作の実装としては、コントローラーの左スティックで前進・後退・向きの調整を行い、右スティックで上下の移動を行えるようにした。対応表を表1に示す。

表1 コントローラーと動作の対応表

| コントローラーの操作 | 対応動作 |
|---|---|
|  |  <p data-bbox="1002 1711 1152 1742">向きの調整</p> |



3.3 距離画像の生成と CG オブジェクトの行動範囲の制限

動作を実装しただけではテクスチャマッピングされた球面の外側にユニティちゃんが行ってしまうなどの不自然な移動が発生し、違和感を感じてしまう。そこで、立体360°画像を使用してあらかじめ距離画像を作成する。この距離画像を使って、ユニティちゃんの行動範囲を制限する。OpenCV3のStereoSGBMクラスを使用し、距離画像を作成した。ステレオマッチング結果を図7に示す。

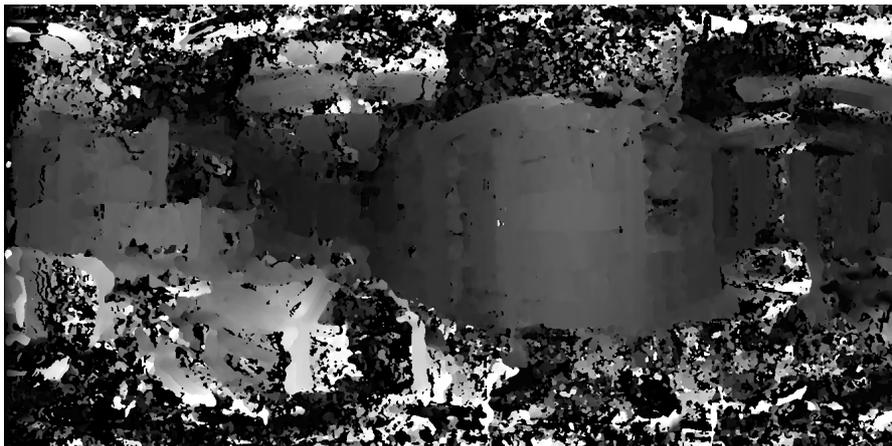


図7 ステレオマッチングの結果

図7のマッチング結果は正確にマッチングできていない部分が多く、距離画像として使用することは難しい。そこでマッチングできていない部分に関しては、付近の濃淡値から推測した濃淡値で補正する作業を行った。また、その際にメガネなどの細かなアイ

テムに関しては考慮しないで補正を行った。補正結果を図8に示す。

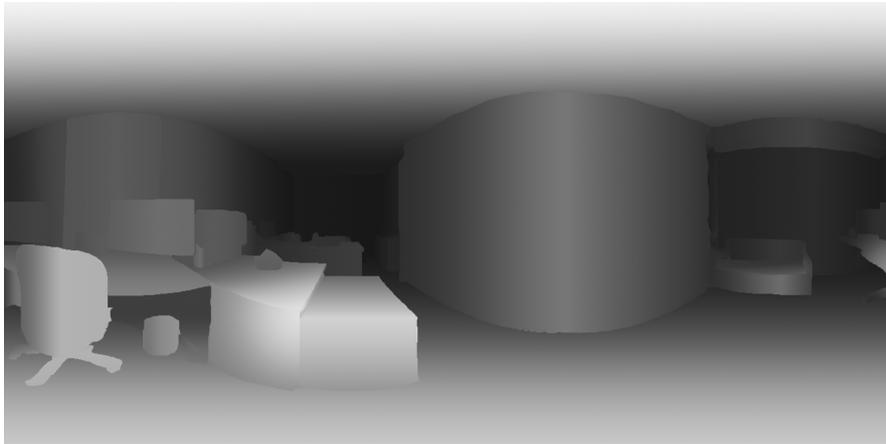


図8 補正後の距離画像

補正後の距離画像を用いてユニティちゃんの移動範囲の制限を実装した。まず事前に、距離画像の画素値（距離値）とカメラからの距離の対応をとった。ユニティちゃんを移動させる場合には、カメラとユニティちゃんの座標点を結んだ直線の延長線上と球面の交わる点に対応する距離値を参照し、その距離よりもカメラから離れる動作が入力されている場合、その動作を無効にした。動作を制限した結果、椅子にぶつかってそれ以上奥にいけない状態と壁際までいけた状態を図9に示す。



(a) 椅子にぶつかっている様子



(b) 壁にぶつかっている様子

図9 移動範囲制限の実装結果

3.4 CG オブジェクトの両眼視差の調整

ユニティちゃんの両眼視差の調整について述べる。人が物体を見ている時、その物体が遠ざかればその物体に対しての両眼視差は小さくなる。物体が近づけば大きくなる。本研究で、ユニティちゃんは、距離画像による制限付きではあるが、空間内を自由に動き回る。従って、両眼視差を距離に応じて変化させることが自然である。ここで、通常

の両眼視と本システムが実装する両眼視の比較を図 10 に表す。図 10(a)のように、通常は単一の物体を見た位置の違いから距離を感じる。本研究で使用する方法では、ユニティちゃんは右目側・左目側と別々に存在する。この違いが両眼視差に与える影響を述べる。

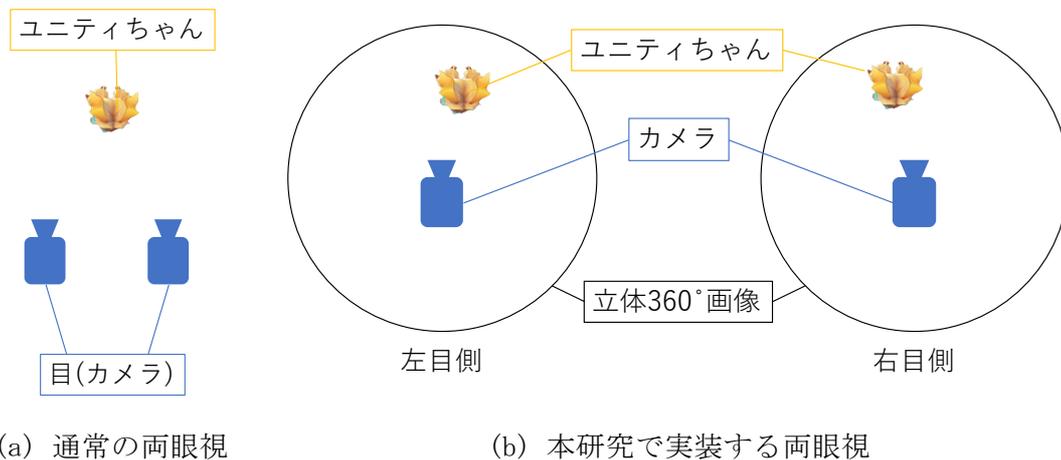


図 10 通常の両眼視と本研究で実装する両眼視

まず、人がある物体 A を正面に見ている状態を図 11(a)に示す。この状態を本研究の両眼視で表すとカメラからの相対的な位置から図 11(b)の物体の配置となる。

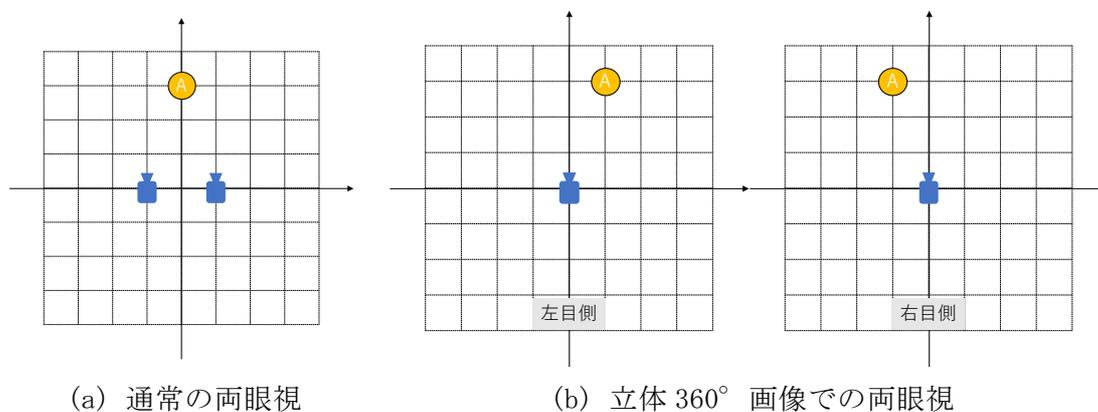


図 11 図 10 に座標系を設定した様子

次に、図 12 のように物体が左に 2 マス・下に 1 マス移動する場合を考える。この場合、カメラからの相対位置を考えると、通常の両眼視と立体 360° 画像の両眼視で、物体の移動量は同一になる。つまり、初めに両眼視差が適切である場合、対象物の移動に関しては通常の方法と本研究の方法の違いを考慮しなくて良いことがわかる。

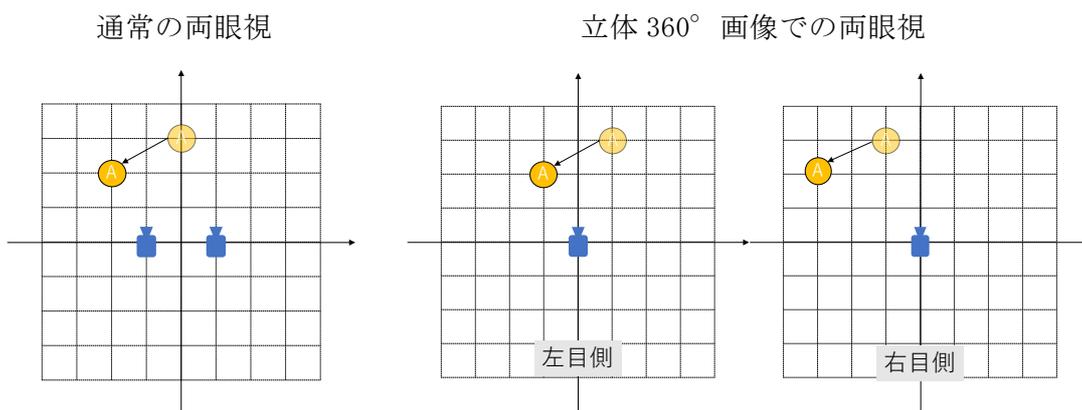


図 12 対象物の移動

では、実際にどのような場合に両眼視差を考慮しなければならないかという、カメラの回転に対する場合である。人が 90° 左を向く場合、(厳密には目の位置は頭を中心より少し前にあるが) 図 13 の左図のようにカメラが図の原点を中心に 90° 回転する形となる。しかし、本研究の環境ではカメラは位置の移動はせず回転はカメラ自体の中心でカメラの角度のみ変わるという結果となる(図 13 の右図)。この場合、カメラと物体の相対距離が変化してしまい、両眼視差がずれてしまう現象が発生する。つまり、頭の回転に応じてユニティちゃんの位置を調整しなければならないということである。

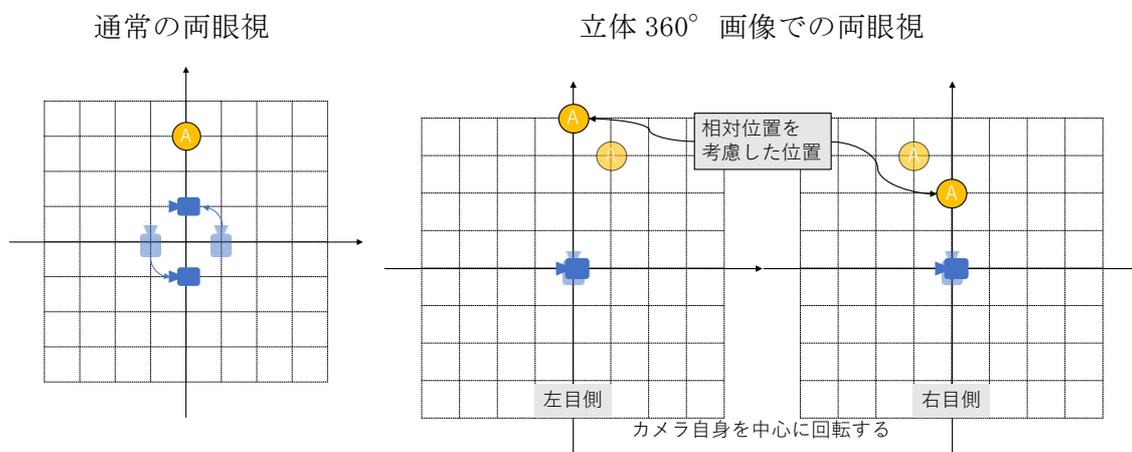


図 13 頭の回転による影響

以上の事から、プレイヤーの頭の回転を参照してユニティちゃんの両眼視差の位置を調整する機能を実装しなければならない事がわかった。

人は、頭を中心を軸として回転すると考えると目の移動は頭を軸とした円の軌道を描くはずである。したがって、動かすべきユニティちゃんも相対的に円の軌道を描いた両眼視差の調整となる。そこで、数学関数の \sin 及び \cos を用いて両眼視差の調整を行う。

円の半径を 0.3 とし，頭の角度を参照し，位置情報を調整したところ，自然な視差が感じられるようになった。

4章 研究結果と考察

立体 360° 画像に 3 次元 CG オブジェクトを重畳合成する場合の、動作の実装と両眼視差の調整を行った。結果、ある程度の違和感を除くことに成功したが、違和感が完全になくなったとは言えなかった。原因として次のことが挙げられる。

- (1) 距離画像の精度が不十分である
- (2) ユニティちゃんの移動にモーションを実装していない
- (3) 運動視差について考慮していない
- (4) 光の影響を考慮していない
- (5) 距離に応じた像の大きさになっているか

今回、距離画像を手動で補正し用いたが設定した濃淡値と実際の両眼視差から感じられる距離感にズレがあり、部分的に不自然な位置関係となる場所ができてしまった。

(2) に関しては、今回人型の CG オブジェクトを採用したために挙げた問題である。やはり人型などの動物系統に関しては動作自体も考慮に入れるべきである。

(3)(4)(5) は、今回両眼視差を実装したが、他にも人が距離感を感じる方法は細かく分ければまだ存在する。その一部が運動視差という対象物が移動している時にその移動する速度から遠近を判断するものや、対象の陰影から位置を判断する、対象の大きさから距離を推定する方法などがある。最も影響が大きいのが両眼視差ではあるが、違和感を取り除くためにはこう言った部分についても、実装できれば良くなると思われる。

5章 結論

実写立体 360°画像にCGオブジェクトを違和感なく合成する手法として両眼視差を実装した。しかし、実写立体 360°画像をテクスチャマッピングしたスフィア内のさらに限られた空間の中しか移動できないなどまだまだ制限が大きい部分に不安要素を感じる。また、4章で述べたように改善・検討すべき点も挙げられている。

しかし、Google のストリートビューのように視点がある程度自由に移動できるような立体 360°画像にこのような合成が可能になれば、建造物がある敷地に建てようとした場合や、部屋内に新しい家具を設置しようとした場合などのシミュレーションなど実用的な部分での、より現実的でわかり易い想定が可能になるだろうと考えられる。

参考文献

[1] Virtualized Reality

<http://www.cs.cmu.edu/~virtualized-reality/>

[2] 日本バーチャルリアリティ学会

<https://vrsj.org>

[3] 横矢 直和(2005) “現実世界と仮想世界を融合する複合現実感技術-I 複合現実感とは”J-STAGE,システム/制御/情報,49(12),pp.489-494

[4] 笠原 凱(2017) “立体 360°画像とナビゲーションに関する研究” 京都産業大学コンピュータ理工学部 特別研究Ⅱ

[5] Paul Milgram and Fumio Kishino (1994)“A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS” IEICE,E77-D(12),

謝辞

本論文を作成にあたり,丁寧な御指導を賜りました蚊野浩教授に感謝いたします.

付録 本研究で開発したプログラムの説明などを付録とする

プログラムの名称，開発環境，言語，仕様，関数の名前と機能などを説明する．プログラムのリストはつけない．

表 2 作成したプログラム

| 名前 | 説明 |
|---------------------------|---|
| ImagePointCalclate.cs | ユニティちゃんとカメラの位置をもとに，対応する距離画像の画素位置を計算する． |
| UnityChanControllLeft.cs | 左目側のユニティちゃんの動作を実装したプログラム．行動範囲の制限もここで実装している．Unity のワールド座標系を参照しているので右目側と左目側で計算が微妙に異なる． |
| UnityChanControllRight.cs | 右目側のユニティちゃんの動作を実装したプログラム． |
| StereoMatching3.cpp | 距離画像を作成するプログラム．Opencv の stereoSGBM クラスを使用しマッチングを行う．さらにマッチングの精度を上げるために，フィルタリングや使用した入力画像のサイズ調整を行うように改良したもの． |