

ユーザ視点に基づく透過型拡張現実インタフェース

See-through Augmented Reality Interface based on User's Point of View

鈴木 伶央^{*1}
Reo SUZUKI

林 佑樹^{*1}
Yuki HAYASHI

中野 有紀子^{*1}
Yukiko NAKANO

^{*1} 成蹊大学理工学部

Faculty of Science and Technology, Seikei University

Augmented reality (AR) expands the real-world environment by adding virtual information to enhance user's perception in the real world. Some studies have proposed AR systems that display virtual objects by video see-through methods. However it requires users to wear special instruments such as a head mount display. In this research, in order to enable more realistic and natural AR interaction in a large room, we propose an AR displaying method based on the user's location. The user's location and poses are detected by Kinect's user-tracking information. Virtual objects are integrally displayed to a large transparent screen according to the user's location and vertical intervals of the environment. We implement two prototype systems in which a virtual object is displayed on the screen, moving between two real world objects, and the virtual object's movement changes according to the user's pose and the position of the real objects.

1. はじめに

現実環境に情報を付加することで現実世界を拡張する拡張現実(AR)に関する研究が盛んに行われている。ビジョンセンサとビデオシースルーを組み合わせた現実物体と仮想物体の前後関係の表現を可能にした AR システム[神原 1999]が提案されている。しかし、ビデオシースルー方式をとる AR では、ビデオ映像に CG が合成されるため、現実世界を肉眼で見るわけではない。また、ユーザは HMD などの装置を装着する必要がある。これに対して、透過型スクリーン越しに仮想オブジェクトに触れることができるシステム[Hilliges 2012]が提案されている。透過型スクリーンを利用することで、実際に現実に重畳されているかのような投影を行うことができる。しかし、インタラクションが行える空間は手が動かせる領域のみであり、ユーザの行動できる範囲が限られてしまうという問題がある。

本研究では、ユーザの位置と動作情報に基づき、仮想オブジェクトを大型スクリーンの背後の現実世界に重畳できる拡張現実インタフェースを提案することを目的とする。リアリティの高い AR を実現するために、スクリーン背後の環境を見ることができる大型の透過型スクリーンを利用する。ユーザの位置と動作の認識には、Kinect のユーザトラッキング機能を利用する。また、Kinect の深度情報に基づき透過型スクリーン背後の空間に置かれた実物体の高さを認識することで、透過型スクリーンに投影している仮想オブジェクトを実物体に合わせて表示できるシステムを構築する。

2. システムの環境構成

図 1 に対象とするシステム環境を示す。想定するユーザは1名であり、ユーザの位置と動作に応じて現実世界に重畳された仮想オブジェクトとのインタラクションを実現する。システムは大きく分けて 4 つの機能から構成される。まず(1)ユーザの位置情報の特定を行う。これはユーザの位置が変化すると物体を見る角度が変わるため、位置を特定して投影するためのカメラの位置に反映させるためである。次に(2)仮想オブジェクトとのインタ

ラクションを実現するためのユーザ動作情報の認識を行う。動作情報は特定のポーズをしたときに認識する形で実現する。ユーザのポーズを認識したときに、仮想オブジェクトの振舞いを変化させる。そして、(3)仮想オブジェクトを投影する実空間の認識を行う。仮想のオブジェクトは投影先の空間である、透過型スクリーンの背後の形状に応じて動作しなければならない。そのために、スクリーン背後の空間を認識する必要がある。最後にこれらの情報を統合し、(4)ユーザの位置に応じた仮想オブジェクトをスクリーンに表示する。

本研究では、(1), (2), (3)の機能を実現するために、深度センサを搭載した Microsoft 社の Kinect¹を 3 台利用する。また、(4)を実現するために、2.0m×1.5m の大型の透過型スクリーンを用いる。システムを構成した部屋の広さは、横 3.5m 縦 5.7m 高さ 3.0m で行う。以下の節で、各機能の詳細を説明する。

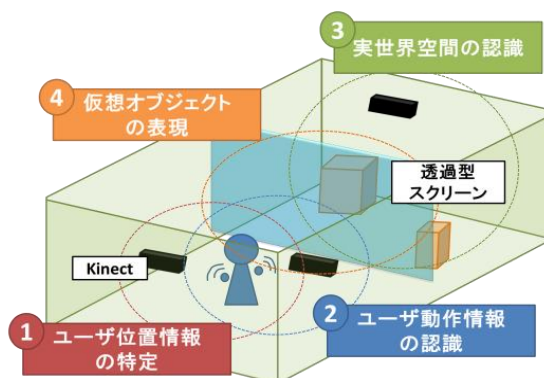


図 1. 想定する環境

2.1 ユーザ位置の特定

オープンソースライブラリとして提供されている OpenNI²は、Kinect から得られる RGB 画像、深度データ、ユーザトラッキングなどの情報を取得できる。本研究では OpenNI のユーザトラッキング機能を利用して、ユーザの位置情報を特定する。ユーザ

連絡先: 鈴木 伶央, 成蹊大学理工学部, 東京都武蔵野市吉祥寺北町 3-3-1, 0422-37-3756, us102055@cc.seikei.ac.jp

¹ Kinect : <http://www.microsoft.com/en-us/kinectforwindows/>

² OpenNI : <http://www.openni.org/>

トラッキングでは図 2 に示される骨格 15 点について、それぞれ三次元座標情報で取得することができる。

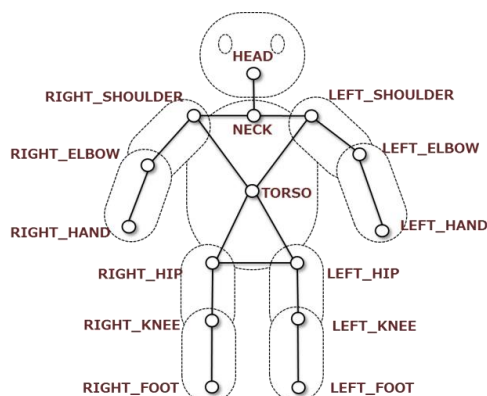


図 2. Kinect で取得できる骨格座標

システムでは、視線位置の高さでユーザ背面の壁に取り付けられた Kinect から、ユーザトラッキングにおける頭部三次元座標座標 (図 2 の HEAD) を取得する。座標は Kinect から距離として取得されるため、Kinect の設置位置を原点とすることで、得られた x, y, z 座標の値をそのままユーザの位置として利用することができる。

2.2 ユーザのポーズ認識

ユーザのポーズを認識するために、Kinect から得られる 15 点の 3 次元座標情報に基づき、認識するポーズを定義する。骨格座標点のうち連続している 2 点からなるベクトルを、26 方向の単位ベクトルで指定する。このベクトルを複数組み合わせることで、ポーズの定義を行う。

ユーザトラッキングから取得している 3 次元骨格座標に基づく骨格ベクトルと、定義したポーズを表す全てのベクトルのコサイン類似度を計算することでポーズを認識する。判定は毎フレーム行う。コサイン類似度が全て 0.9 以上 (誤差範囲 10° 程度) であれば一致したポーズをしていると判定する。ポーズは、予め任意の数を xml 形式で登録することができる。

ポーズの定義例を図 3 に示す。この例で定義されたポーズは左手を左に、右足を右斜めにしたポーズであり (図 3 左)、赤くなっている 2 点間が xml で指定するベクトルである。図 3 右側がユーザトラッキングの様子である。赤い円で囲ってある骨格と、定義された骨格のベクトル方向が一致しているかが計算され、全て一致していればポーズが認識される。

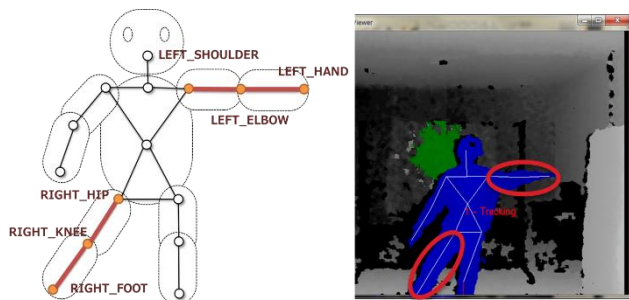


図 3. ポーズの定義例

2.3 投影対象世界の空間認識

仮想オブジェクトを投影する実空間を認識するために、スクリーン奥側の空間に高さ 2.9m の位置に下向きに設置された Kinect の深度情報を利用する。Kinect は 640×480 の解像度で各ピクセルの深度を得ることができる。計算量を削減するために、本研究では解像度を標準の 1/5 倍にあたる 128×96 の深度に解像度を落としている。深度情報は Kinect から距離として取得されているため、設置されている床からの高さを引くことで、各ピクセルにどれくらいの高さの物体があるか把握できる。また、Kinect から 2.7m 以上離れている部分は深度情報を取得しない。これは計算量の削減と、高さの誤認識を防ぐためである。

図 4 に投影対象世界を Kinect の深度センサで捉えた様子を示す。右側が実際に取得した深度情報を等高線グラフで表した図である。高さが取得できている部分は緑から赤で表し、それ以外の未取得の部分は青で表してある。物体がある位置の床からの高さが取得されていることがわかる。対象となる実世界に置かれたオブジェクトを認識することで、高低差に応じた仮想オブジェクトをスクリーンに投影することが可能となる。

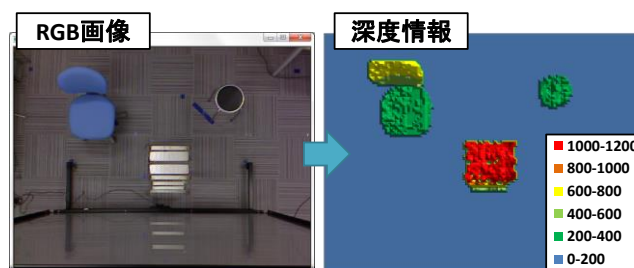


図 4. 投影対象世界の深度画像

2.4 透過型スクリーンへのオブジェクト投影

3 次元のオブジェクトをスクリーンに表示するために、グラフィックス処理のための API として提供されている OpenGL³ を用いる。投影には平行投影と透視投影の 2 種類があるが、平行投影では遠近感を表現することができないため、glFrustum 関数を用いて透視投影を行った。

図 5 に、glFrustum で設定可能な描画範囲を示す。視線方向を中心に、Near とその距離にあるクリップ面の Top, Bottom, Left, Right の端までの距離と、奥行きを表す Far の 6 つを引数に設定できる。この 6 つの引数から実際に描画が行われる範囲である視錐台の大きさが決まる。ここで引数に与える値には、現実世界のサイズと同じ値を用いる。また、視線方向は投影面に垂直となるように固定し、ユーザ位置から計算される透過型スクリーンの位置と常に一致させるように、引数の値を更新する。Kinect で取得したユーザ位置を OpenGL 内のカメラ座標とし、スクリーンに仮想オブジェクトを投影する。これにより、ユーザ位置に応じた透視投影が実現する。

³ OpenGL : <http://www.opengl.org/>

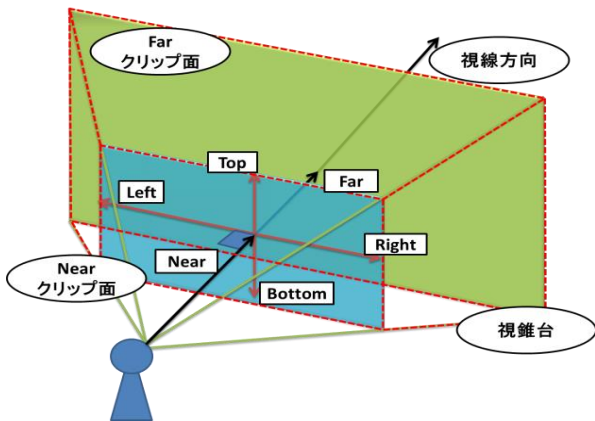


図 5. glFrustum の描画範囲

3. システム構成

2.1～2.4 節で述べた機能を統合したシステムを構築した。図 6 にシステムの構成図を示す。本システムは C++及び Java 言語を利用して実装している。システムは、4 つのモジュール(ユーザ位置認識, ポーズ認識, 空間認識, 描画処理)から構成されており, ソケット通信で必要なデータを送受信する。

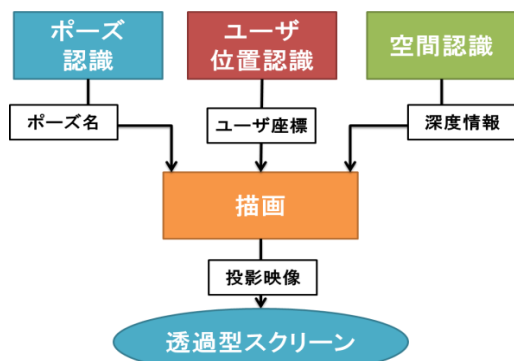


図 6. システム構成図

ユーザ位置認識モジュールでは, ユーザ後方に設置された Kinect から頭部までの距離を送信する。ポーズ認識モジュールでは, ユーザ前面に配置された Kinect より, ポーズ認識時にそのポーズ名を送信する。今回定義したポーズは図 7 に示した 4 つである。A, B は手を左右に, C では右手を右斜め上に, D は左手を左斜め上に右足を右斜めにしたポーズである。空間認識モジュールでは, 天井に設置された Kinect から, 深度情報を格納した配列を送信する。描画モジュールでは, 3 つのモジュールから送られてきた各種情報に基づき, 透過型スクリーンに投影する映像を生成し, プロジェクタで投影する。

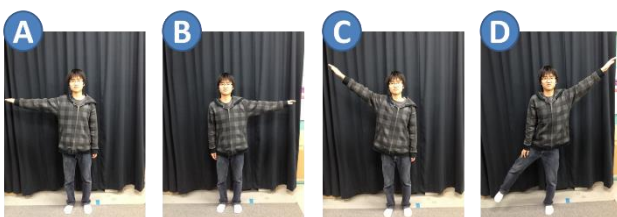


図 7. 定義したポーズの一覧

4. システムの表示例

本システムを用いて 2 つのインタラクションを達成できるシステムを実装した。1 つは, 仮想オブジェクトが実世界に置かれた椅子の間を往復するものである。図 7 の C のポーズをすると, 仮想オブジェクトが左右に動く。画面外に達するか, 仮想オブジェクトの高さよりも高い実物体が進行方法にある場合, 反対側へ進行を変える。

図 8 にシステムの様子を示す。①から③は目線の位置のカメラから撮影した映像である。ポーズが認識されると, ①のように移動を開始し, ②では椅子が進行方向にあるため, 逆方向へ進行方向を変える。③では再び椅子があるため, また逆方向へ進行を行う。

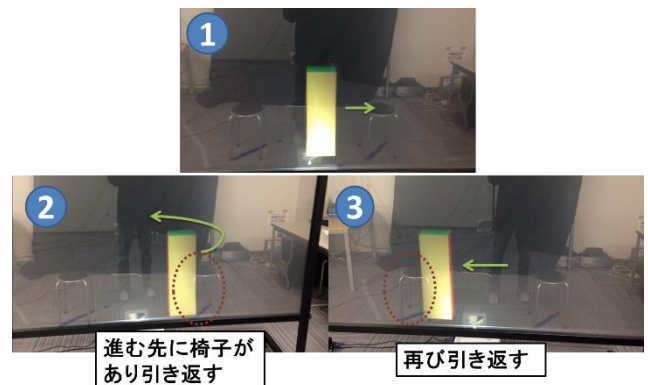


図 8. システム 1 の表示

もう 1 つは, 現実の物体上から別の物体上に仮想オブジェクトが飛び移るというものである。図 7 の D のポーズをすると, 高さがある実物体の上に移動する。移動する際は, 仮想オブジェクトが乗っている実物体以外に移動する。空間認識はリアルタイムで処理されており, 投影対象世界の実オブジェクトの配置の移動や追加に応じて, 仮想オブジェクトのアニメーションも変化する。

図 9 に 2 つ目のシステムを実行した様子を示す。①から⑤は目線の位置にあるカメラからの映像である。①で, 最初に右側にある椅子の上に仮想オブジェクトが乗っている。そこで図 7 の D のポーズを認識すると, ②のように別の実物体として検出されている椅子への移動を開始する。そして③では左側の椅子の上に移動が完了している。また, 途中で物体を追加した場合でも認識が行われ, ポーズをすると④から⑤のように追加した椅子の上に仮想物体が移動する。

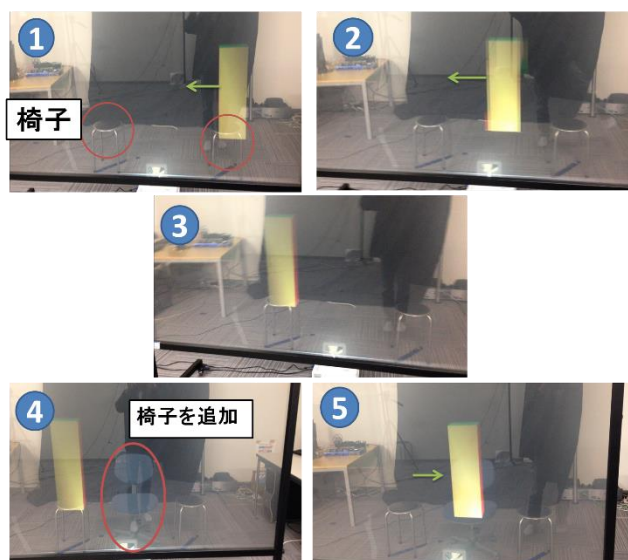


図 9.システム 2 の表示例

5. おわりに

本研究では、ユーザ位置や動きに応じて、透過型スクリーンの背後の現実世界と、スクリーンに提示された CG を重畳投影するインタフェースを提案した。また、構築した 2 つのシステムでユーザとのインタラクションを達成した。ユーザ位置による仮想オブジェクトの見え方の変化、ユーザの動作によって高さがある物体の間を往復と、高さがある物体の上を飛び移るシステムである。

今後の課題として、透過型拡張現実インタフェースを基盤とし、物理演算を導入して仮想オブジェクトの複雑な動きや実物体との接触判定を取り入れていく。また、ユーザのポーズ認識だけでなく、ジェスチャなどのユーザの動作に応じたインタラクションを実現できる環境へと拡張していく予定である。

参考文献

- [神原 1999] 神原ら:「ビデオスルー型拡張現実感のための実時間ステレオ画像合成」, 電子情報通信学会論文誌, Vol.J82-DII, No.10, pp.1-9 (1999).
- [Hilliges 2012] Hilliges, O. et al.: "HoloDesk: Direct 3D Interactions with a Situated See-Through Display", Proc. of CHI'12, pp.2421-2430 (2012).