

Pinch-n-Paste: AR環境におけるテクスチャ転写用 ジェスチャインタラクションシステム

森 磨美¹ 間下以大² 清川 清² 竹村治雄²

概要: 現実空間で物体の複製や模様の変更などの操作を行うことは困難である。提案システムではそれらの操作を Augmented Reality(AR) 空間上で素手によるインタラクションによって実現する。RGB-D カメラを用いてモデルベースの手形状認識を行い、関節の3次元位置を取得する。また各関節の角度を特徴ベクトルとして、隠れマルコフモデルによって学習、状態推定を行う。提案手法では手を広げた状態の Normal, 人差し指と親指を伸ばした Pinch, 全ての指を曲げた Grab の3つの状態を認識し、オブジェクトに触れた手の状態によって、テクスチャの転写, オブジェクトの移動, オブジェクトの複製を実行する。実験により、ジェスチャの認識を行いジェスチャに対応した機能を実行できることを確認した。

1. 序論

従来の AR システムでは、あらかじめ用意されたコンテンツ以外を利用することができず、また AR を体験しながら新しい AR コンテンツを作ったり、既存のコンテンツを編集するといったことも困難であった。

我々の研究室では、AR ジオラマ [1] という 3D 復元システムの研究が行われてきた。これらのシステムでは、ユーザは現実物体を復元し、新しいバーチャル物体として AR 環境に重畳表示することができる。また、復元した物体の形状やテクスチャをインタラクティブに変更するための研究として、馬勝らの Pinch-n-Paste[2] がある。

Pinch-n-Paste は AR 空間上でテクスチャ転写を行うシステムである。ユーザは入力装置を使わずに手や指の動きによってテクスチャを物体から取り出し、他の物体に貼り付けることができる。この研究では、復元したモデルとテクスチャを用いて現実物体の外見を変えることを目的としている。例えば家具や服の模様を変えることが可能となり、インテリアデザインやファッションの分野で利用することができる。しかし、この Pinch-n-Paste ではテクスチャ転写のみが可能で、他のコピーなどの編集機能がない。Pinch-n-Paste をデザインなどの用途に実際に使用するためには、テクスチャ転写だけでなく、物体の移動やコピーなどの編集機能も必要である。また表示装置はモニタのみとなっているため、AR 空間での位置の把握が困難になっていると考えられる。

本報告では、この Pinch-n-Paste を改良し、テクスチャ転写だけでなく、物体の移動、コピーも行えるシステムを提案する。これらの機能を追加することで、部屋の椅子を 3D モデルで復元し、複製をして好きな位置に配置し、それぞれ好きなテクスチャを貼り付ける等の操作が可能になる。これらの機能を追加するために手認識の手法を変更し、モデルベースの手形状認識と隠れマルコフモデルを用いてハンドジェスチャを認識できるようにした。また、シースルー Head Mounted Display(HMD) を利用し、視覚と位置覚の一致を行った。

以下、2 節では先行研究となる馬勝らの Pinch-n-Paste の概要と問題点について述べ、ジェスチャ、手認識に関する関連研究について述べる。3 節では本研究のシステム的设计方針について述べ、4 節で実装手法について述べる。5 節では実験の結果、6 節で結論と今後の課題について述べる。

2. 関連研究

2.1 Pinch-n-Paste

馬勝らの Pinch-n-Paste は、AR テクスチャ転写システムであり、AR 空間上にあるオブジェクトに素手で直接接触れることでそのオブジェクトからテクスチャを取り出し、他のオブジェクトへテクスチャを転写することができる。(図 1) またシステムに利用するオブジェクトは現実物体を復元したものを利用することができ、実空間の物体を AR 空間上で復元し編集することが可能となっている。

2.1.1 問題点

馬勝らの Pinch-n-Paste で考えられる問題点について述

¹ 大阪大学 大学院情報科学研究科

² 大阪大学 サイバーメディアセンター

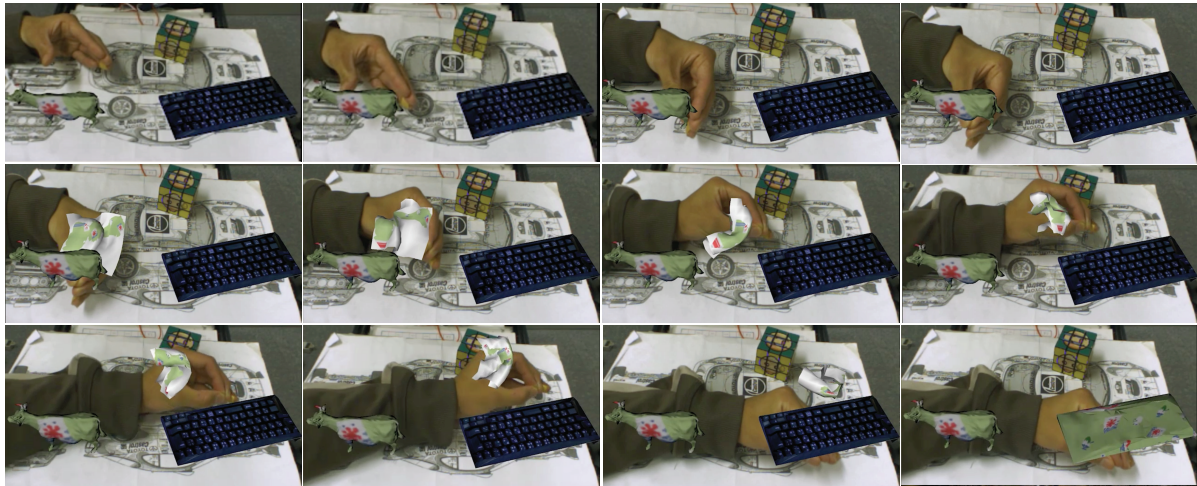


図 1 馬勝らの Pinch-n-Paste.

べる.

実物体の三次元形状の獲得

実物体の三次元形状の獲得は、テクスチャ転写インタラクションとは実際には切り離されており、復元をリアルタイムに行いながらテクスチャ転写を行うことができない。

テクスチャ転写

他の物体から貼り付けられたテクスチャは物体の展開図に合わせて貼り付けられるが、ペーストしたテクスチャにひずみが生じることがあり、元の物体の特徴が現れにくい。

素手によるインタラクション

肌色の領域を検出し、その深度によって AR 空間上にバーチャルハンドを生成している。肌色の判定は環境の光源や人による肌の色の違いに影響を受けるため、安定して領域を検出することができない。また、バーチャルハンドは指先の検出は行うが、手の形状についての認識は行わないため、ジェスチャの認識を行うことができない。現在はオブジェクトに接触するという操作方法しかなく、そのため機能もテクスチャを取り出し貼り付けるだけとなっている。

表示装置

表示装置は図 2 のようにモニターとなっており、モニターを見ながら手を動かす必要がある。カメラの視点とユーザの視点の位置が一致せず、自然な姿勢での作業ができず疲労感につながると考えられる。またカメラの位置を自由に動かせるようにした場合、ユーザがインタラクションに利用できるのは片手だけに限られる。

2.2 ジェスチャに関する研究

Epps らのテーブルトップジェスチャインタラクションで使われる手形状の研究 [3] では、テーブルトップジェス

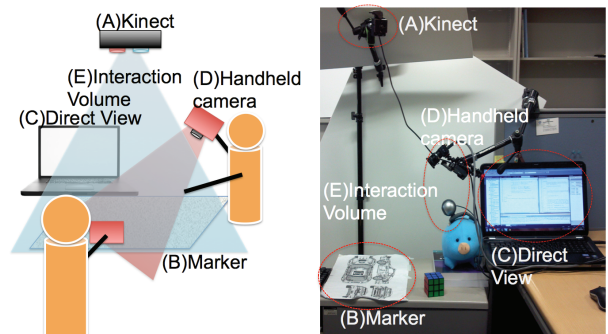


図 2 馬勝らの Pinch-n-Paste のハードウェアセッティング.

チャインタラクションシステムで採用されている手形状と機能の対応についてまとめられている。

2.3 手認識に関する研究

Lee らのナチュラルハンドインタラクションの研究 [4] では、RGB 画像から肌色領域を切り出し、手のひらの中心と指先を検出することで指の向きを計算し、レイキャスティングによって AR 空間上のオブジェクトとのインタラクションを行う。

Hilliges らの HoloDesk[5] は、3D ダイレクトインタラクションシステムである。ーフミラーを利用してバーチャルオブジェクトの表示を行い、検出した手や実物体とバーチャルオブジェクトとの物理シミュレーションによりオブジェクトをすくったり挿入したりすることが可能である。

Wang らのマーカレスハンドトラッキングの研究 [6] では、カメラを 2 つ用いることで違う角度からの手のシルエットを得るため、カメラが 1 つの場合よりも正確に手形状の認識を行うことができる。

Oikonomidis らのモデルベース 3D ハンドトラッキング [7] は、Kinect を利用して手形状の認識を行い、隠れた部分の認識も可能となっている。手形状の認識には粒子群

最適化を利用している。

羽岡らの隠れマルコフモデルに基づくハンドジェスチャーアニメーション作成 [8] では、CyberGlove を用いて取得した手形状から、各関節の回転角度をパラメータとして隠れマルコフモデルの学習を行い、学習サンプルにはない指定したジェスチャーの系列のアニメーションの生成を行う。

2.4 本研究の位置づけ

本研究では、馬勝らの Pinch-n-Paste の問題点の中で、素手によるインタラクションと表示装置の問題の解決を検討する。モデルベースの手形状認識に変更し、それにより得た関節の 3 次元位置の情報からジェスチャーの認識を行う。ナチュラルハンドインタラクションのみで AR 空間上のオブジェクトの直接操作を可能にする。また表示装置としてシースルー HMD を利用可能とし、視覚・位置覚の一致を行う。

3. システム設計

本節では、本研究のシステムの設計方針について述べる。

3.1 ジェスチャーによる機能の追加

追加する機能の決定と、それに対応したジェスチャーの決定の手法について述べる。

3.1.1 追加機能の検討

本システムに最低限必要と思われる機能として以下の 2 つを選択した。

- オブジェクトの移動
- オブジェクトのコピー

馬勝らの Pinch-n-Paste ではオブジェクトの位置は固定されており、自由に移動させることができなかった。インテリアなどのデザインを考える際に、オブジェクトを自由な位置に配置させて検討することは重要なことである。また、オブジェクトの複製を可能にすることで、実空間では検証しにくい実物体の複数配置を実現することができる。

3.1.2 ジェスチャーの決定

追加する機能を実現するために採用するジェスチャーの手形状を決定する。Epps らのテーブルトップジェスチャーインタラクションで使われる手形状の研究 [3] を参考に、よく使われているジェスチャーの中から機能に対して直感的に使えるものを選択する。

テクスチャーの転写は、実空間で物体の表面から皮をはがすときの動作に近い Pinch を採用する。具体的には人差し指と親指のみを伸ばし、他の指は曲げた状態を指す。この状態でオブジェクトに触れるとオブジェクトからテクスチャーを取り出すことができる。

オブジェクトの移動は、実空間で物体を移動するときの動作に近い Grab を採用する。具体的にはすべての指を曲げている状態を指す。この状態でオブジェクトに触れると

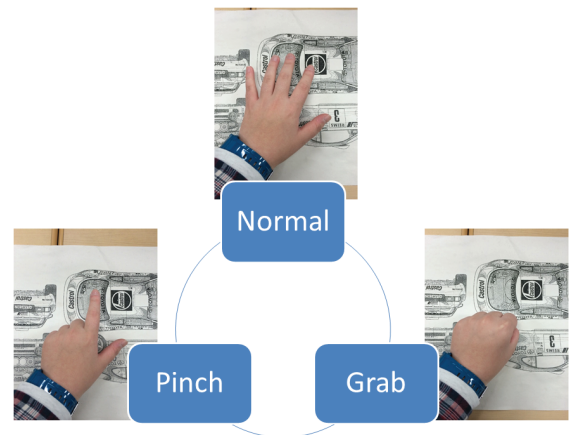


図 3 認識する 3 つの手の状態。

オブジェクトの位置を移動することができ、Grab の状態から通常の状態に戻るとオブジェクトをその位置に置くことができる。

オブジェクトのコピーは、実空間で実際に行うことはできないが、粘土を二つに分割するときのように、両手で Grab を作りオブジェクトに触れるという動作によって実現する。

よって、認識すべきジェスチャーは Pinch と Grab になり、この 2 つと通常状態の合わせて 3 つの状態を認識できるようにする。それぞれの状態は図 3 のようになる。

3.2 手認識の高精度化

手認識の高精度化手法とジェスチャー認識の手法について述べる。

3.2.1 形状認識

本研究ではモデルベースの手認識手法を採用した。モデルベースの手認識手法では、手の表面だけでなく関節の情報も利用することができるため、指を曲げているかどうかなどのジェスチャー認識に利用することができる。Wang らのマーカレスハンドトラッキングの研究 [6] によると、RGB カメラによって得た手領域のシルエットにより手形状の認識が可能となり、また深度カメラを用いることにより深度値による背景と手領域のセグメンテーションが可能になる。

3.2.2 ジェスチャー認識

馬勝らの Pinch-n-Paste では、認識した指先とオブジェクトとの衝突判定のみでジェスチャーを認識しており、機能の追加などの拡張性に問題があった。

ジェスチャー認識の手法として、隠れマルコフモデルが一般的に利用されている。隠れマルコフモデルとは観測可能なパラメータから未知のパラメータを推測する確率モデルであり、時系列パターン認識に向いている。今回実装するジェスチャーは手で形を作るのみだが、動きのあるジェスチャーを採用する際に容易に拡張ができるため本システムで

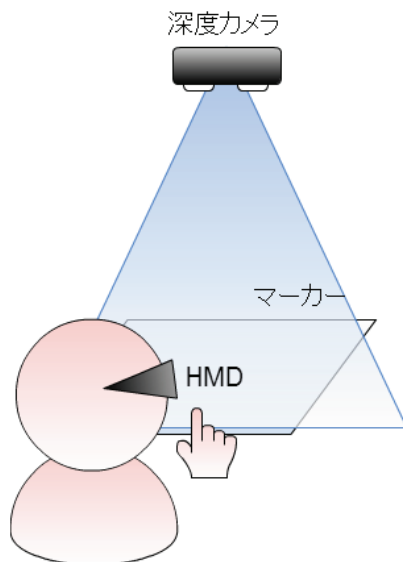


図 4 ハードウェアセットアップ.

は隠れマルコフモデルを利用する.

前項で得た手形状のモデルより各指の関節の角度や指先の位置を得ることができる. 本システムではそれらの値を特徴ベクトルとして隠れマルコフモデルによって学習, 認識を行う.

3.3 視覚・位置覚の一致

システムの視覚・位置覚を一致させるために, シースルー HMD を用いる. シースルー HMD はカメラが 2 つ付いているものを用いる. そのカメラによって得た映像に AR コンテンツを重畳表示させ, サイドバイサイド方式により立体視を可能とする. これにより, ユーザの視点とカメラの視点を一致させることができ, また立体視により位置の把握が容易になると考えられる.

4. システム実装

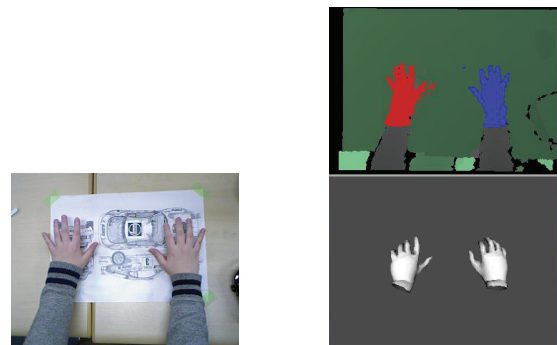
本節ではシステムの実装手法について述べる.

4.1 ハードウェア構成

システムは図 4 のように, マーカーの真上に深度カメラを設置し, ユーザは HMD を着用してマーカーの前に座り, マーカー上に手を伸ばすことでオブジェクトの操作を行う.

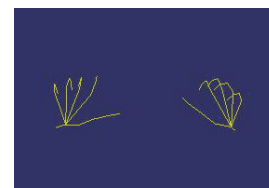
4.2 システム構成

システムの構成は図 5 のようになっており, サーバで深度カメラにより手形状認識を行い, 状態推定や衝突判定によりオブジェクトの情報を編集する. サーバ・クライアント間ではオブジェクトやバーチャルハンドの位置情報について通信を行い, クライアント側では受信した位置情報をもとに, HMD から得た映像に対してレンダリングを行う.



(a) 深度カメラの RGB 画像.

(b) 3 Gear Systems.



(c) 取得した関節情報.

図 6 3 Gear Systems を用いた関節情報の取得.

4.3 手認識の高精度化

手認識の高精度化のための実装手法について述べる.

4.3.1 モデルマッチングによる形状認識

手形状を取得するために, 3 Gear Systems[9] というシステムを取り入れた. 3 Gear Systems は, Kinect などの RGB-D カメラを利用してモデルベースにより手の姿勢の認識を行い, 図 6 のように, 深度カメラの深度画像から手の領域をセグメントして手の形状を復元し, そこから手の関節のデータを得ることができるシステムである. この関節の座標ごとに球を作り, バーチャルハンドを生成する. 深度カメラは馬勝らの Pinch-n-Paste システムと同じく作業領域の真上に配置するため, 作業環境を変更することなく導入することが可能である.

3 Gear Systems ではハンドトラッキングサーバを実行し, そのサーバと通信を行うことで認識した手の各関節の三次元位置などの情報を 3 Gear Systems の世界座標系で得ることができる. これに, 深度カメラ座標系から世界座標系への変換行列 W_T の逆行列 W_T^{-1} を掛けることで深度カメラ座標に変換し, システムに用いる. (図 7 参照)

4.3.2 隠れマルコフモデルによるジェスチャ認識

3 Gear Systems によって得た関節の座標データを元に隠れマルコフモデルを用いてジェスチャの認識を行う.

モデルの学習には Baum-Welch アルゴリズムを使用する. 学習にはあらかじめ取得した関節データを元に目視で正解の状態のラベル付を行い, 状態ごとに特徴ベクトルの

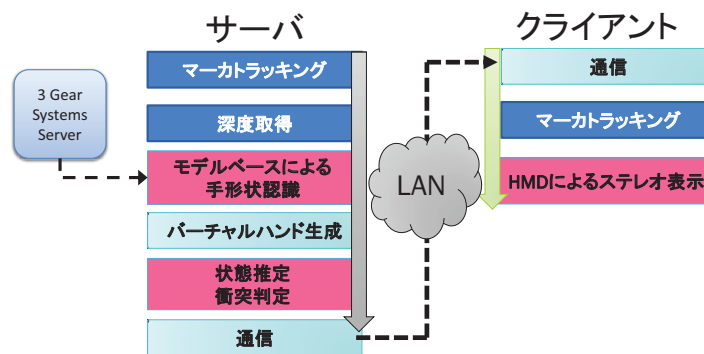


図 5 システムの構成.

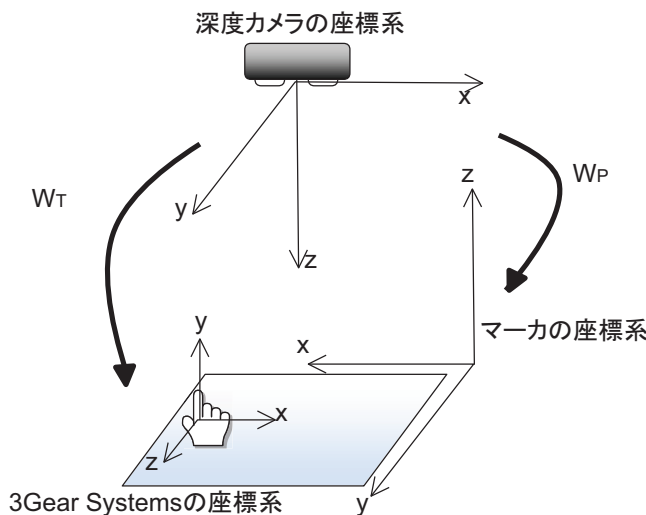


図 7 座標系の変換.

平均ベクトル, 分散共分散行列, 状態遷移確率を計算し, その数値を初期値として学習を行った.

特徴ベクトルとして関節の角度の大きさを使用した. このパラメータを用いることで, 手の向きや位置などに影響を受けず, 指を曲げるという動作を認識しやすくなると期待できる.

学習を行ったモデルで, サンプルデータを用いて Viterbi アルゴリズムによる推定を行ったところ表 1 のような結果となった.

3 Gear Systems によって関節の座標情報を得るたびに,

表 1 学習の結果.

正解率	F 値 (Normal, Pinch, Grab)	F 値の平均
0.836	0.929, 0.644, 0.740	0.770

特徴ベクトルとなる関節の角度の計算を行い, 観測系列に加えて Viterbi アルゴリズムによって推定される状態の系列を計算する. その結果の系列の中で一番最後のものを現在の状態とする.

隠れマルコフモデルの実装は General Hidden Markov Model(GHMM) ライブラリ [10] を用いた Python プログラムによって行った. システムの子プロセスでこの Python プログラムを実行し, パイプにより座標情報と推定した情報のやり取りを行うことで状態の推定を行う.

4.4 ジェスチャによる機能の追加

3 Gear Systems によって生成したバーチャルハンドとオブジェクトとの衝突判定と, 隠れマルコフモデルによる状態推定によって取得した現在の手の状態によって図 8 のような状態遷移を行う. 図中で説明のない矢印は無条件の遷移を表す. これらの状態遷移はサーバ PC 上で行う.

テクスチャ転写は手が Pinch の状態の時にオブジェクトに衝突した場合に行われる. Pinch の状態でオブジェクトに衝突すると, システムはオブジェクトのテクスチャを取り出し, ハンカチ状のテクスチャオブジェクトを生成する. その後, そのテクスチャオブジェクトを AR 空間上のバーチャルハンドの位置に合わせて移動させ, 他のオブジェクトに衝突した時にテクスチャの貼り付けを行い, 初期状態へ戻る. (図 9)

オブジェクトの移動は手が Grab の状態の時にオブジェクトに衝突した場合に行われる. Grab が保たれている間, 対象のオブジェクトをバーチャルハンドの位置に移動させる. 両手とも Grab 以外の状態になった場合, オブジェクトの移動を終了し, その後初期状態へ戻る. (図 10)

オブジェクトのコピーは両手が Grab の状態で同じオブジェクトに衝突した場合に行われる. オブジェクトの複製

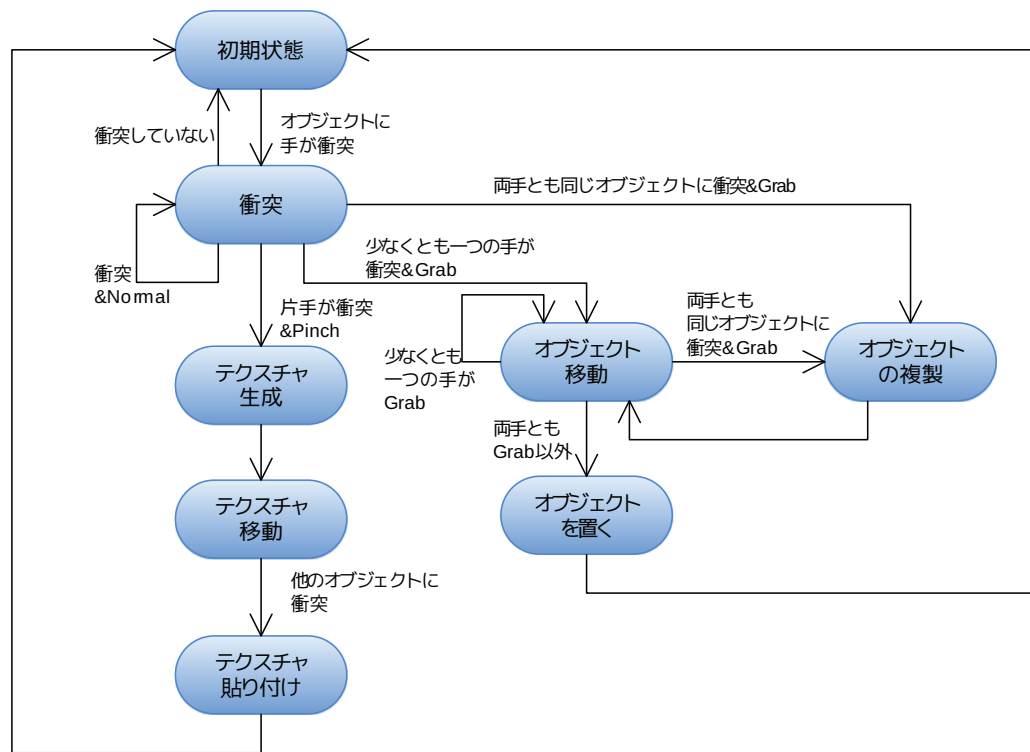


図 8 ジェスチャ認識の状態遷移.

を行った後はただちにオブジェクトの移動に遷移しそれぞれの手で複製したオブジェクトの移動を行うことができる。またオブジェクトの移動の間に両手とも Grab で同じオブジェクトに衝突した場合にもコピーを行う。(図 11)

4.5 HMD による視覚・位置覚の一致

HMD のカメラから得る映像は USB2.0 の帯域の中で 30FPS で安定して使うために 320×240 とし、左右のカメラからそれぞれキャプチャを取得し、AR コンテンツを重ねて画面に出力する。マーカトラッキングは、左右独立に行うと位置推定誤差のためにコンテンツの両眼立体視が不安定になったため、右のカメラのみで行う。マーカトラッキングにより得た変換行列 P_R と一定量の平行移動行列 T_M から、左のカメラの変換行列 P_L を式 (1) によって計算する。

$$P_L = T_M P_R \quad (1)$$

各変換行列によってレンダリングされた AR コンテンツを、カメラからのキャプチャを重ねて図 12 のようにサイドバイサイド方式で表示し、HMD での立体視を可能とする。

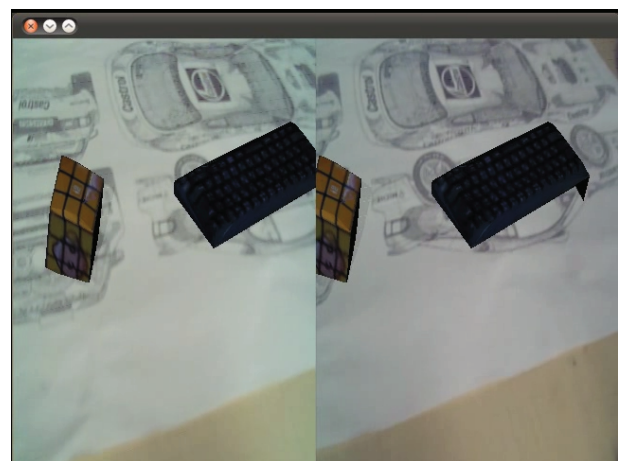


図 12 サイドバイサイド方式による表示.

5. 実験

本節では、実装したシステムの評価実験に関して述べる。実験 A ではジェスチャ認識の精度の評価を行う。実験 B ではシステム全体の評価を行う。実験は被験者 8 人に対して行った。

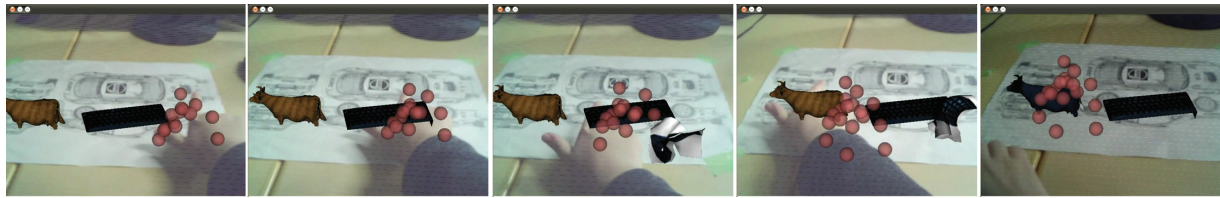


図 9 テクスチャ転写.

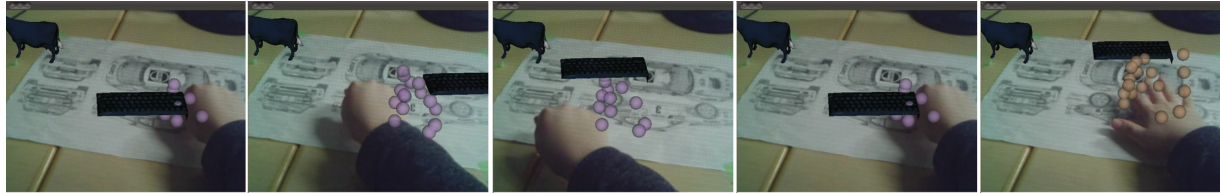


図 10 オブジェクトの移動.

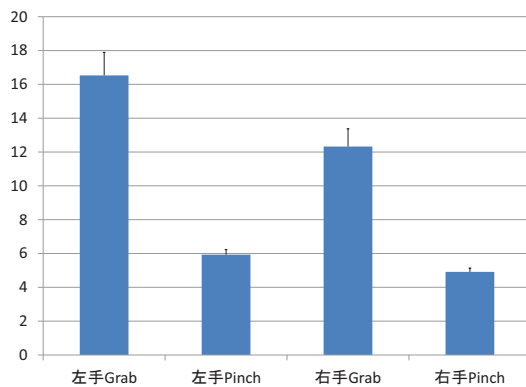


図 13 ジェスチャ認識の平均時間 (秒).

5.1 実験 A. ジェスチャ認識の精度評価

実験 A では実装したジェスチャ認識の精度評価を行う。

5.1.1 実験設定

Normal 状態から指定したジェスチャへ手の状態を変更し、ジェスチャが正しく認識されるまでの時間を計測する。以下の条件で実験を行った。

手 { 右手, 左手 }

ジェスチャ { Pinch, Paste }

試行回数 20

5.1.2 実験結果

被験者に対して認識対象となるジェスチャを掲示してから、ジェスチャが認識されるまでの平均時間は図 13 のようになった。関節の状態と推定された状態を比較し、混同行列を作成し再現率、適合率、F 値の計算を行った結果、表 2 のようになり、F 値の平均は 77.6%であった。

5.1.3 考察

Grab の状態を Pinch と誤認識することが多く、正しく認識されるまでに時間がかかった。誤認識の原因は 3 Gear Systems により得た関節情報では手の握りが緩く認識されることが多いためと、学習段階の正解のラベル付を RGB

画像ではなく、関節の 3 次元位置から再現したスケルトンによって行ったためと考えられる。また、3 Gear Systems からの関節情報の取得や、サーバ側のデバッグ用表示へのレンダリングなどの処理に時間がかかり、平均で 3.68FPS のシステム処理速度になっているため、認識時間が大きくなった。

5.2 実験 B. システム全体の評価

システムの機能を被験者に体験してもらい、アンケートによって評価を行った。

5.2.1 実験設定

実験では被験者は HMD を着用しシステムの機能の体験をする。あらかじめ AR 空間上にオブジェクトを表示させておき、そのオブジェクトに対してテクスチャ転写、オブジェクトの移動、オブジェクトのコピーを行う。

5.2.2 実験結果

アンケートは 5 段階のリッツカート尺度により行い、その平均は図 14 のようになった。また、アンケートでは追加すべき機能として以下の機能が挙げられた。

- ひとつ前に戻る, 進む
- 全体を初期化
- オブジェクトの拡大・縮小
- 物体の切り取り, 結合
- テクスチャの保存
- テクスチャー一覧の表示

5.2.3 考察

実験 A で明らかになったように、ジェスチャの認識率の低さと処理速度の遅さを被験者が体感していることが明

表 2 ジェスチャ認識の結果.

	再現率 (%)	適合率 (%)	F 値 (%)
Normal	97.6	95.3	96.4
Pinch	97.4	48.4	64.7
Grab	56.1	99.1	71.6

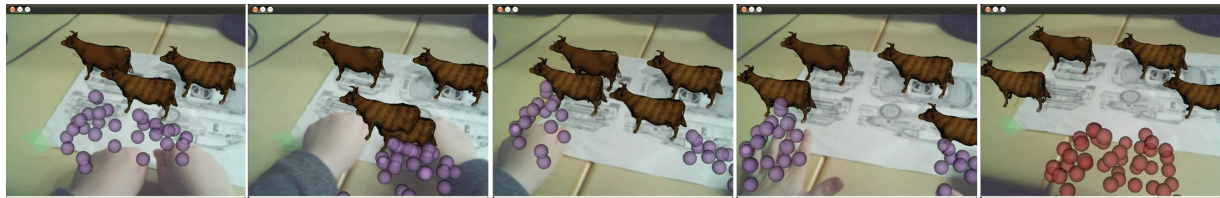


図 11 オブジェクトのコピー.

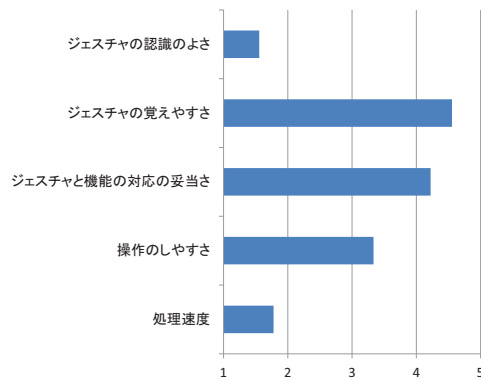


図 14 アンケートの結果.

らかになった。だが、ジェスチャと機能の対応の妥当さやジェスチャの覚えやすさについては高い評価を得ているため、ジェスチャの認識精度の向上を行うことでより評価の高いシステムになると考えられる。また、追加すべき機能にはオブジェクトへの操作だけでなく、テキストへの操作やシステム全体への操作もあげられていた。

6. 結論

本研究では、モデルベースの手形状認識を行うことで安定したバーチャルハンドの生成と、それによって得た関節の情報を特徴ベクトルとして隠れマルコフモデルによるジェスチャ認識を行った。手のジェスチャ認識と従来のオブジェクトとバーチャルハンドの衝突判定を組み合わせることで、テキスト転写だけでなく、オブジェクトの移動とコピーの機能を実現した。ジェスチャ認識の認識精度は結果的にあまり高くなく、正しいジェスチャが認識されるまでに時間がかかる。また、システム自体の処理速度も遅いためバーチャルハンドの表示にもずれが生じることがあった。

シースルー HMD による視覚・位置覚の一致も行った。本手法ではシースルー HMD の左右のカメラから映像を取得し、AR コンテンツを重ねてシースルー HMD に表示する。

今後の課題として、ジェスチャ認識の誤認識率の削減、各モジュールの別スレッド化によるシステムの処理速度の向上、オブジェクトの回転や変形、テキストの保存や再利用などの今回実装を見送った機能やアンケートにより挙

げられた追加の機能の実装などが挙げられる。また、実物体の復元をリアルタイムに行い、現実空間にある実物体に対しても様々な機能によって操作できるようにする必要がある。

謝辞 本研究の一部は、文部科学省科学研究費補助金(課題番号 24300048)の補助による。

参考文献

- [1] Umakatsu, A., Yasuhara, H., Mashita, T., Kiyokawa, K. and Takemura, H.: Camera-Based In-situ 3D Modeling Techniques for AR Diorama in Ubiquitous Virtual Reality., *Proc. Int. Conf. on Human-Computer Interaction (HCI International)*, Vol. 6773, pp. 113-122 (2011).
- [2] Umakatsu, A., Mashita, T., Kiyokawa, K. and Takemura, H.: Pinch-n-Paste: Direct Texture Transfer Interaction in Augmented Reality, *IEEE Virtual Reality (VR)*, pp. 73-74 (2013).
- [3] Epps, J., Lichman, S. and Wu, M.: A Study of Hand Shape Use in Tabletop Gesture Interaction, *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pp. 748-753 (2006).
- [4] Lee, M., Green, R. and Billingham, M.: 3D Natural Hand Interaction for AR Applications, *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pp. 1-6 (2008).
- [5] Hilliges, O., Kim, D., Izadi, S., Weiss, M. and Wilson, A.: HoloDesk: Direct 3D Interactions with a Situated See-through Display, *Proceedings of the Special Interest Group on Computer-Human Interaction (SIGCHI) Conference on Human Factors in Computing Systems*, pp. 2421-2430 (2012).
- [6] Wang, R., Paris, S. and Popović, J.: 6D Hands: Markerless Hand-tracking for Computer Aided Design, *Proceedings of the 24th Annual Association for Computing Machinery (ACM) Symposium on User Interface Software and Technology*, pp. 549-558 (2011).
- [7] Oikonomidis, I., Kyriazis, N. and Argyros, A.: Efficient model-based 3D tracking of hand articulations using Kinect, *Proceedings of the British Machine Vision Conference*, pp. 101.1-101.11 (2011).
- [8] 羽岡哲郎, 益子貴史, 小林隆夫: 隠れマルコフモデルに基づくハンドジェスチャーアニメーション生成, 電子情報通信学会技術研究報告. CS, 通信方式, Vol. 102, No. 517, pp. 43-48 (2002).
- [9] Wang, R., Twigg, C. and Kin, K.: 3 Gear Systems, <http://www.threegear.com/>, (最終アクセス日: 2014/4/16) .
- [10] Schliep, A.: GHMM Library, <http://ghmm.org/>, (最終アクセス日: 2014/4/16) .