

Leap Motion Controller を用いた AR 物体の把持動作表示システムの構築 Development of A System enabling Display of Gripping the AR Object Using Leap Motion Controller

片平 怜士*¹
Reiji Katahira

曾我 真人*²
Masato Soga

*¹ 和歌山大学大学院システム工学研究科
Graduate School of Engineering, Wakayama University

*² 和歌山大学システム工学部
Faculty of Systems Engineering, Wakayama University

Abstract: Augmented Reality (AR) in the existing systems has a problem that the drawn objects are always displayed in the foreground because 3D models by AR are superimposed later than the picture of the actual world. In this study, we proposed a system to produce a realistic picture of AR in accordance with every depth. We developed a prototype system to verify the effect of the method. The prototype system was developed by focusing on a human hand. We utilized Leap Motion Controller as a motion capture device to acquire the depth data of the hand and fingers.

1. はじめに

1.1 研究背景

仮想空間と現実空間とを融合する Augmented Reality (以降 AR) の技術は, ARToolKit の登場により, 周知の技術となりつつある[加藤 02]. しかし, 現実空間の映像にあとから重畳されるため, AR によって重ね表示される 3D モデルは常に最前面に表示されてしまい, 不自然な情景となる場合があった.

現在, この問題に対してさまざまなアプローチで解決に向けての研究がなされている.

2012 年に Silvio R. R. Sanches らが発表したシステムでは, 3D モデルを表示するためのマーカ以外に, 視点からの距離を測るために, 別途マーカを準備していた[Silvio 12]. このマーカを人の胸部に貼り付け, 身体的位置を認識させて空間上の前後関係を計算した. しかし, 単一のカメラで複数のマーカを同一視野内で同時に認識しなければならないため, カメラとマーカとの距離を一定以上に保つ必要があり, また, 認識状態が保てるような, 大きさのあるマーカが必要だった.

2013 年に Dong Woo Seo らが発表したシステムでは, Kinect の深度カメラを用いた画像処理により, 映像から手の部分を抽出し, 最前面に描画するように表示域を調整していた[Dong 13]. しかし, AR によって描画された 3D モデルに対して身体のある部分は隠れて, ある部分は前面に表示される動作には対応していなかった.

1.2 研究目的

本研究では, 1.1 節であげた問題点を踏まえ, 深度に応じた自然な情景を表示可能な AR 描画手法を提案し, その試作システムの構築を行う. 構築するシステムは, 試作段階として手の動作, その中でも把持動作に着目して構築する. 手は日常生活の中で最も使用頻度の高い部位であるため試作段階として採用した. なお, 対象動作を把持動作としているため, 指先以外の体の部位は考慮しない.

システムの動作目標として AR により表示される物体より奥にある指は見えない状態にし, 手前にある指は見える状態にする(図 1 参照).

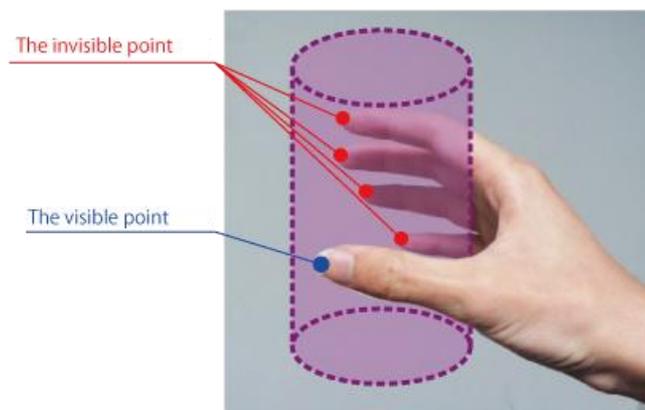


図 1 手法による目的とする挙動

2. 提案手法

本研究では, 透過した指の 3D モデルを使用し, ユーザの各指に追従させる. この透過した指の 3D モデルをマーカ上に表示する物体の 3D モデルよりも先に描画することで, 後から描画しても範囲内は影響を受けない[尾川 10](図 2 参照). よって, 1.2 節であげた問題点にあった, 複数のマーカを同時に認識させる必要がなく, また, マーカの位置を動かすことなく, 仮想モデルと実空間との距離関係を測ることができる.

指の 3D モデルを手指に追従させるためには, ユーザの指の位置情報を正確に取得しなければならない. 本研究では, 正確な指位置を取得するために, 専用のモーションキャプチャ機器を使用する. 今回は物体への把持動作に対応させるため, モーションキャプチャ機器として, 手指に特化している Leap Motion Controller を採用する. このデバイスを用いて正確な指の位置情報を取得し, 実際の動作に合わせて 3D モデルの位置・向きをユーザの手指に対応させる.

連絡先: 片平 怜士, 和歌山大学大学院システム工学研究科
我研究室, 〒640-8510 和歌山県和歌山市栄谷 930,
TEL:073-457-8457, {s165012, soga}@center.wakayama-u.ac.jp

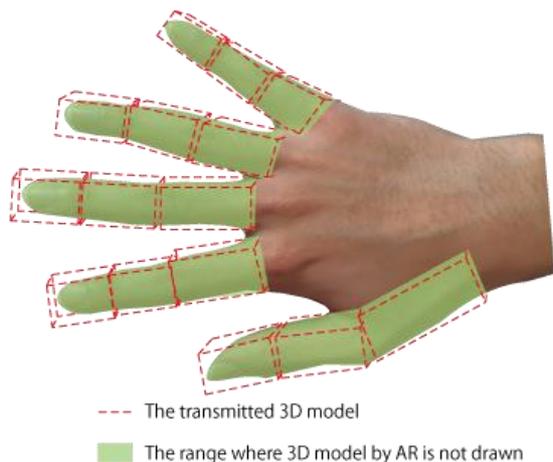


図 2 影響を受けない範囲

3. 開発環境

本システムを構成するハードウェアは以下である。

- PC
- Leap Motion Controller
- Web カメラ
- AR 用のマーカ

これらを組み合わせ、システムの動作する環境を設定する。このとき、Leap Motion Controller とマーカは互いの設置位置を考慮にいれなければならない。Leap Motion Controller の座標系と AR マーカの座標系の中心位置がズレるためである。この Leap Motion Controller とマーカの位置関係を 4.2 節の座標変換時に使用する(図 3 参照)。

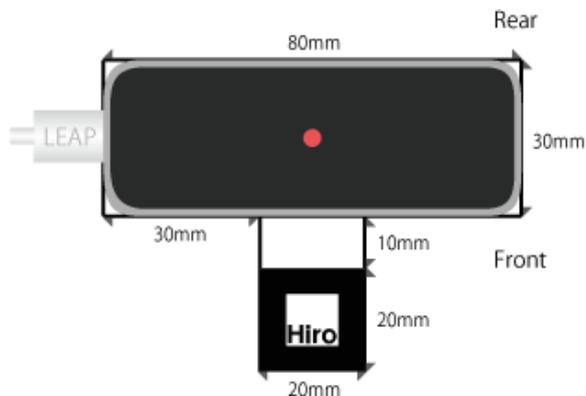


図 3 Leap Motion Controller と AR マーカの位置関係

4. 処理概要

本システムでは一連の流れの中で、主に次の処理を行っている。

- 初期化処理
- 3D モデルの読み込み
- 座標変換
- 描画処理

これらの処理を順に実行していくことで、目的の挙動を行う。

4.1 初期化処理

本システムでは、各ライブラリおよびパラメータの初期化以外に、処理を行う前に設定しておくものを既定のステータス情報と

して初期化処理の中に格納している。格納しておくステータス情報はマーカ・3D モデルに個々に設定する固有データと、マーカ・3D モデルに共通して設定する共通データがある。

固有データには、パターンファイル名、パターン ID、マーカ ID、3D モデル名、.mqo ファイル名がある。このうち、パターン ID は初期化処理の中で主にビットマップデータなどのマーカに関する情報を取得し、個別に割り当てる。マーカ ID は自然数を識別番号として割り当てる。

共有データには、可視状態の検出フラグ、パターンの中心座標、パターンの幅がある。このうち可視状態の検出フラグは初期状態で不可視に設定しておく。また、パターンの中心座標は X・Y の両座標とも 0 に設定しておく。

4.2 3D モデルの読み込み

本システムでは、3D モデリングソフトである、Metasequoia を用いてマーカ上および指に追従させる指の 3D モデルを作成している。この 3D モデルの情報を読み込むために Metasequoia の保存形式であるテキストファイル、.mqo ファイルを解析し、頂点数や頂点座標、マテリアルといった必要な情報を保存してデータを扱っている。

4.3 座標変換

本システムでは、把持動作への対応をするために Leap Motion Controller からさまざまな情報を取得している。取得した座標値の座標系と OpenGL の座標系は異なるため、座標変換を行う必要がある(図 4 参照)。

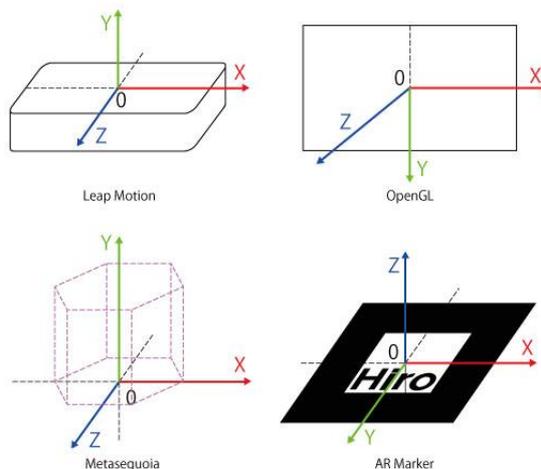


図 4 各座標系

図 3 のように Leap Motion Controller と AR マーカを配置するため、座標系の中心位置を考慮に入れる必要がある。Leap Motion Controller で取得した座標値を L とし(式 1 参照)、中心位置の離れた距離を仮に X 軸方向に a mm, Y 軸方向に b mm, Z 軸方向に c mm とすると、変換後の座標値 L' は次のようになる(式 2 参照)。

$$L = \begin{bmatrix} leap_x \\ leap_y \\ leap_z \end{bmatrix} \dots\dots\dots (1)$$

$$L' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} leap_x \\ leap_y \\ leap_z \end{bmatrix} + \begin{bmatrix} a \\ b \\ c \end{bmatrix} \\ = \begin{bmatrix} leap_x + a \\ -leap_z + b \\ leap_y + c \end{bmatrix} \dots\dots\dots (2)$$

また, Metasequoia で作成した 3D モデルをマーカー上に表示するためにもそれぞれ座標系が異なるため座標変換を行う必要がある。

4.4 描画処理

Leap Motion Controller で取得した手指のデータを元に, 透過した指の 3D モデルを描画する。描画関数は ARToolKit にて準備されたものを用いており, 描画する位置は指の各骨の中心位置である。透過した指の 3D モデルをマーカー上に表示する 3D モデルよりも先に描画することで, 後からマーカー上の 3D モデルについて描画の記述をしても範囲内は影響を受けない。これにより目的の挙動を行うことができる(図 5 参照)。

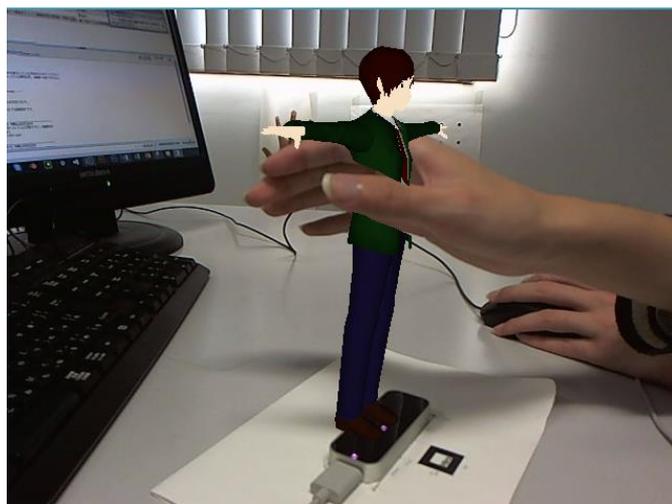


図 5 描画した情景

5. 検証実験

本システムを構築後, 検証実験を 2 つ行った。検証実験 I ではマーカー上に表示する 3D モデルの種類によってシステムの挙動に変化がないか確認した。検証実験 II では被験者を募りアンケート調査を行った。

5.1 検証実験 I

構築したシステムが, マーカー上に描画する 3D モデルのさまざまな種類に対応可能かを検証する。また, 3D モデルの種類に加えて, ユーザが行う動作の種類も把持動作とは別に数パターン用意し, 正しい挙動をしているか各モデルごとに検証する。

(1) 評価方法

3D モデル 15 パターンに対して(図 6 参照), 動作 6 パターンをそれぞれ 5 回ずつ行う。5 回動作を行った結果, 5 回正常に動作を行えた場合は○, 2 回以上 4 回以下の場合は△, 1 回未満の場合は×として記録をつける。○の場合は+1, △場合は 0. ×の場合は-1としてまとめる。数値の合計値から今回構築したシステムが対応している動作を確認する。

本システムが安定に挙動する 3D モデルパターンおよび, 動作パターンの傾向を明らかにする。検証した動作は以下の 6 パターンである(動作詳細は表 1 参照)。

- (ア) 横からつかむ
- (イ) 上からつかむ
- (ウ) 手を開いてつかむ
- (エ) 手を閉じてつかむ
- (オ) 人差し指と中指ではさむ
- (カ) 横から手を差し込む

(ア)~(オ)は一度 Leap Motion Controller にデバイスの直上で手を認識させてから動作を行う。(カ)は Leap Motion Controller に手を認識させず, 手を開いた状態でセンサの範囲外から差し入れる。

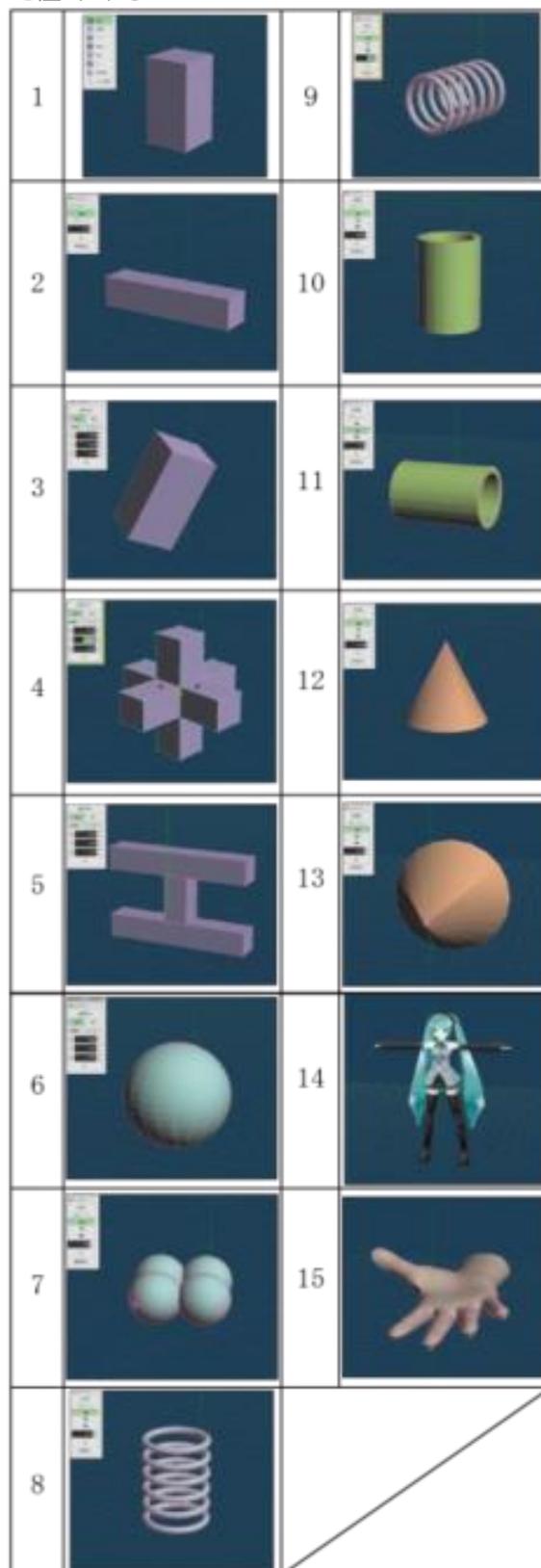


図 6 検証実験 I に用いた 3D モデル
(1~15 の数字は 3D モデルの ID)

表1 検証実験 I で設定した動作詳細

(ID)	動作詳細
(ア)	手をセンサに認識させた後、物体を横からつかむように手を丸める。
(イ)	手をセンサに認識させた後、物体を上からつかむように手を丸める。
(ウ)	手を高い位置でセンサに認識させ、そのままセンサ方向に下ろす。
(エ)	手を高い位置でセンサに認識させ、指先を丸めた状態でセンサ方向に下ろす。
(オ)	手をセンサに認識させ、物体を人差し指と中指で挟むように手を丸めて動かす。 (チョコキの形状)
(カ)	はじめに認識を行わず、横から手を開いた状態で差し入れる。

(2) 実験結果

把持動作(ア)を行った結果、準備した 3D モデル 15 パターンのうち 14 パターンに対して正常に動作した。このことから、本システムは把持動作に対応していることがわかる(表2参照)。また、把持動作以外の他の動作 5 パターンについても同じ検証を行い、数値の合計からシステムが対応している動作を抽出した。ほとんど動作しないことを表す×が存在しなかったことにより、3D モデルの種類によってシステムの挙動に大きく影響を及ぼさないことがわかった。

また、「指を丸める」といった手指の関節位置がセンサから見えなくなるような動作(イ)(エ)(オ)は総じて合計点数が低く、不安定な結果となった。Leap Motion Controllerでは、認識する点のデータが欠損した場合、欠損するまでの過程から補完処理を行っている。しかし、欠損している時間が長くなると、処理に失敗する可能性がある。よって、動作(イ)(エ)(オ)が不安定になったと考えられる。

以上のことから、本研究で構築したシステムは3Dモデルの種類による挙動の差はないが、動作の種類によってシステムの挙動に影響が出ることがわかった。

表2 検証実験 I の結果
(1~15の数字は3DモデルのID)

Model		1	2	3	4	5	6	7	8
Result	ア	○	○	○	△	○	○	○	○
	イ	△	○	△	△	△	△	△	△
	ウ	○	○	○	○	○	○	○	○
	エ	△	△	△	△	△	△	△	△
	オ	○	○	○	○	○	△	△	△
	カ	○	○	○	○	○	○	○	○
		9	10	11	12	13	14	15	計
	ア	○	○	○	○	○	○	○	14
	イ	△	△	△	△	△	△	△	1
	ウ	○	○	○	○	○	○	○	15
	エ	△	△	△	△	△	△	△	0
	オ	△	△	△	△	△	△	△	5
	カ	○	○	○	○	○	△	○	14

5.2 検証実験 II

構築したシステムの改善点を抽出し、システムの課題を明確にする。

(1) 評価方法

被験者 11 人に既存のシステムと今回構築したシステムの両方を試用してもらい、自由記述形式のアンケートによりシステムの改善点を抽出した。

また、被験者が試用中に行った動作を記録し、その中でも正常に動作しなかった動作について考察し、課題を明確にする。

(2) 実験結果と考察

実験の結果、2つの改善点が抽出された。

- 透過範囲が指の大きさよりも広がっており、空間ができてしまう問題の解消。
- 指位置を長時間認識できない動作への対応
- 手がマーカ上のモデルに触れる・近づくなどしたときに、動的なフィードバックを返す機能の追加

今後これらの問題点をシステムの改善点として考慮し、次の段階のシステムへと繋げていく。

6. まとめ

既存の AR による描画システムには問題点があり、さまざまなアプローチで問題解決に向けての開発がなされてきた。先行研究では、AR で 3D オブジェクトを表示するためのマーカ以外に距離関係の把握のため、別途マーカを準備していた。

本研究ではこの問題点を解決するため、深度に応じた自然な情景を表示可能な AR 描画手法を提案し、その試作システムの設計と構築を行った。試作段階のシステムとして対象動作を把持動作とし、透過した指の 3D モデルを各指に追従させることで目的のシステムを構築した。手指の位置情報を取得するために、手指に特化したモーションキャプチャ機器である Leap Motion Controller を使用した。

システム完成後、検証実験を2つ行った。検証実験 I では構築したシステムが把持動作に対応しており、かつさまざまな 3D モデルに対応しているかの確認と動作の種類によってシステムの挙動に影響があるかを確認した。検証実験 II では構築したシステムを数人の被験者に試用してもらい、アンケート調査により改善点を抽出し、その解決策を考察した。

参考文献

- [加藤 02] 加藤博一: 拡張現実感システム構築ツール ARToolKit の開発, 電子情報通信学会技術研究報告, 2002.
- [Silvio 12] Silvio R. R. Sanches, Daniel M. Tokunaga, Valdinei F. Silva, Antonio C. Sementille, and Romero Tori: Mutual Occlusion between Real and Virtual Elements in Augmented Reality based on Fiducial Markers, Applications of Computer Vision (WACV), 2012 IEEE Workshop on , 2012.
- [Dong 13] Dong Woo Seo, and Jae Yeol Lee: Direct hand touchable interactions in augmented reality environments for natural and intuitive user experiences, Expert Systems with Applications, Volume 40, 2013.
- [尾川 10] 尾川景子, 綿森道夫: 3DCG と Web カメラの合成を表現するプログラムの制作, 高知工科大学電子・光システム工学科, 2010 年度卒業論文, 2010.