



リアルワールドメタバース実現に向けた クロスバース・プラットフォームの構築 (3) —天井設置カメラを用いた複数人物の頭部位置推定機構—

岩崎勇斗¹⁾, 中村文彦²⁾, 木村朝子²⁾, 柴田史久²⁾

1) 立命館大学大学院 情報理工学研究科 (〒 567-8570 大阪府茨木市岩倉町 2-150)

2) 立命館大学 情報理工学部 (〒 567-8570 大阪府茨木市岩倉町 2-150)

概要: 我々は、現実世界とメタバースを融合したリアルワールドメタバースの実現を目指し、現実世界と仮想世界を相互に行き来できるクロスバース・プラットフォームの構築に取り組んでいる。本研究では、実空間における複数の人物による混雑状況を VR 空間に反映するため、人物の頭部位置をリアルタイムで検出するシステムを開発した。複数のネットワークカメラと YOLOv5s により頭部を検出し、その 3 次元位置を VR 空間のアバタに反映させることで、遠隔地にいる参加者に現場の臨場感と混雑状況を提示する。

キーワード: クロスバース・プラットフォーム, メタバース, MR, VR, AR, 物体検出, 位置推定

1. はじめに

近年、複合現実 (MR) 環境におけるデジタルツインの活用が注目を集めている。特に Niantic 社が提唱する「リアルワールドメタバース」は、現実世界をデジタルツインとして VR 空間に再現し、現実とリモートの双方からのアクセスを可能にする新たな概念である。このアプローチは、物理空間と MR 空間のデジタル情報を緊密に連携させることで、遠隔地のユーザーに高い臨場感を提供し、エンターテインメントからビジネスまで広範な応用が期待されている [1]。

こうした背景を踏まえ、我々はリアルワールドメタバース具現化の一環として、大学キャンパスを VR 空間に再現する「MR キャンパス」を構想・提案している [2]。デジタルツインを介した現実空間と VR 空間の相互作用に焦点を当て、現地ユーザーは MR 技術を介して遠隔ユーザーのアバタと対話し、遠隔ユーザーは VR 空間内でアバタとして行動しつつ、現地ユーザーの位置・姿勢が反映された存在とコミュニケーションをとる。これにより、物理的に離れたユーザー同士がシームレスなコラボレーション環境で共同作業を行うことを目指している。

MR キャンパス構想の実現にあたり、本研究では遠隔ユーザーのソーシャルプレゼンスを高める上で特に重要な要素である「現地空間の混雑状況の可視化」に着目した。従来の対面でのコミュニケーションの維持やコストの観点から、現地にいるすべてのユーザーに HMD を装着させることは現実的ではない。そこで天井に見下ろす形で設置したカメラを用いて HMD を装着していないユーザー（以後、非装着者とする）の頭部位置を検出し、その情報を VR 空間上にアバタとしてリアルタイムに反映するシステムの構築を目指す。

2. 天井カメラによる混雑状況の可視化

MR キャンパス実現の課題の 1 つとして、「現地空間の混雑状況を VR 空間に可視化する」ということが挙げられる。混雑状況を VR 空間に反映してキャンパスの状況を詳細に伝えることで、遠隔参加者とのコミュニケーション活性化が期待できる。

しかしながら、現地側に存在するユーザー全員に HMD を装着させることは、利便性やコストの観点から現実的ではない。多数の HMD を準備・管理するコストに加え、日常的なキャンパス活動において HMD の装着を強制することは、ユーザーの利便性を著しく損なう可能性がある。

そこで本研究では、天井カメラを用いて HMD 非装着者の頭部位置を検出し、3 次元座標を推定して VR 空間にアバタとして反映させるシステムを提案する。このアプローチにより、現地ユーザーは通常の活動を継続しながら、その存在と位置情報がリアルタイムで VR 空間に反映される。実現には大人数かつリアルタイム性を持った処理が要求され、キャンパス内の混雑環境では数十人から数百人規模の人物を同時に検出・追跡し、遅延を最小限に抑えた高速処理が不可欠である。

本システムの活用場面として、教授や発表者など詳細な動作の反映が必要な人物には HMD を装着してもらい、身体の関節情報まで含めた高精度なモーションキャプチャを行う。一方、聴講する学生についてはそこまでの詳細な情報は必要ないケースが多い。このような状況で、本システムは天井カメラによる頭部位置検出のみで十分な情報提供を実現し、現地とリモート参加者間のシームレスな空間共有を可能にする。これにより、コストと利便性のバランスを保ちながら、効果的な MR キャンパス環境を構築できる。

3. システム構築

図 1 にシステムの全体構成を示す。システムは (1) ネットワークカメラによる映像取得部、(2) 頭部検出・座標計算を行うサーバ部、(3) アバター表示を行うクライアント部で構成される。カメラは一般的な RGB カメラのみで構築可能とし、混雑感の提示を主目的とするため、精度よりもリアルタイム性を重視した設計とした。

3.1 映像取得とネットワーク通信

映像取得部では、RTSP(Real Time Streaming Protocol)を用いて複数のネットワークカメラから映像のストリーミングを制御する。本システムでは、PoE(Power over Ethernet)対応のネットワークカメラを使用した。サーバ側では別スレッドで映像取得を行い、カメラ台数分の配列に格納して並列処理を実行している。

3.2 YOLOv5s による頭部検出

頭部検出には高速物体検出アルゴリズム YOLOv5s を採用した。YOLOv5s は、YOLOv5 の軽量版であり、高速な物体検出を実現し、リアルタイム性を重視する本システムに適している。Roboflow Universe で公開されている約 1,200 枚の頭部画像データセットで 100 エポックの学習を実施し、評価データセット (113 枚, 150 インスタンス) に対して IoU 閾値 0.5 で mAP 95.5% を達成した。図 2 に YOLOv5s による頭部検出の実行例を示す。

3.3 高さ推定

単眼 RGB 映像から頭部の高さを推定するため、以下の 2 つの手法を比較した。アプローチ 1 は単眼深度推定アルゴリズムである Depth Anything[3] によって算出した相対深度を絶対深度に変換し、床から頭部までの高さを算出する手法で、画角端と中央の床における深度値の差から単位深度当たりの絶対深度値を計算した。アプローチ 2 は日本人の平均頭部サイズ (直径 17.5cm) を仮定し、検出されたバウンディングボックスの左右端の x 座標を x_0, x_1 ($x_0 < x_1$) として、それぞれに対応する視角 θ_{x0}, θ_{x1} を以下のように算出した。図 3 に頭部サイズ仮定による高さ推定の概念図を示す。

$$\theta_x = \frac{\text{image_width}/2 - x}{\text{image_width}} \times \theta \quad (1)$$

ここで、 θ はカメラの水平視野角、image_width は画像の

横幅 (ピクセル数) である。平行線と線分の比の定理から、カメラの設置位置の高さ h と推定したい頭部の高さ H の関係は

$$(h - H) : h = 17.5 : h(\tan(\theta_{x1}) - \tan(\theta_{x0})) \quad (2)$$

となる。

精度検証の結果、アプローチ 1 では平均 18cm の誤差に対し、アプローチ 2 では平均 5cm の誤差に抑えられたため、本システムではアプローチ 2 を採用した。

3.4 実世界座標への変換

画像上のピクセル座標から実世界の 3 次元座標への変換は、MR キャンパスにおける現実空間と VR 空間の位置整合に不可欠である。この変換にはカメラの内部パラメータ (カメラ行列 K と歪み係数) が必要となる。本システムでは、OpenCV の `calibrateCamera` 関数とチェスボードパターンを用いたキャリブレーションにより、焦点距離 (f_x, f_y)、主点 (c_x, c_y)、歪み係数 (k_1, k_2, p_1, p_2, k_3) を事前に取得した。カメラの設置位置は固定であることから、基本的にこの歪みパラメータは一定である。そのため、このパラメータも一度求めた後は、その他カメラパラメータと同様、予め与えられた情報として扱うこととする。

検出された頭部のバウンディングボックスの中心を頭部位置とし、その位置に対して、求めた歪みパラメータ情報をもとに、OpenCV の `undistortPoints` 関数を使用して歪み補正を行う。その後、歪み補正を行った頭部の座標に対して、前節で求めた高さ情報と、カメラの内部パラメータ (焦点距離、画像中心) を用いて座標変換を行う。具体的には、歪み補正後のピクセル座標をカメラの焦点距離と主点を算出した後、歪み補正済みの画像座標 (x, y) と高さ H を用いて、カメラ座標系での 3 次元位置 (X, Y, Z) を計算する。この変換により、画像上のピクセル座標から実世界スケールでの 3 次元座標が得られる。ここに、予め与えられたカメラの外部パラメータの分だけ回転や座標を移動させることで、絶対座標に変換している。

3.5 通信プロトコルとクライアント実装

サーバ・クライアント間の通信には WebSocket プロトコルを採用した。サーバ側は Python の FastAPI フレームワークを使用し、ASGI 準拠の非同期処理により高い並行性

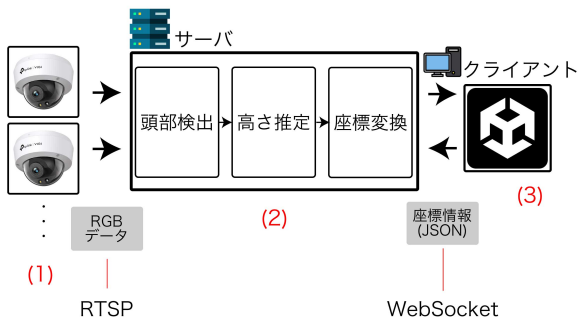


図 1: システムの全体構成

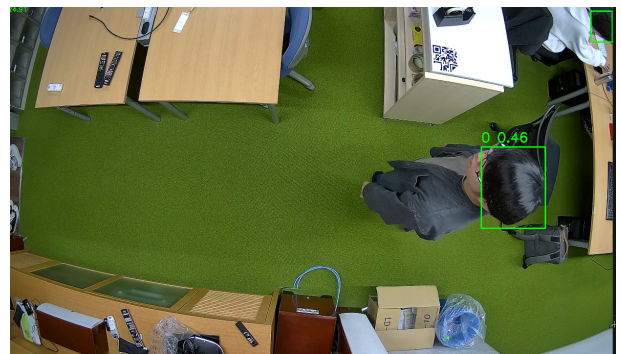


図 2: YOLOv5s による頭部検出の実行例

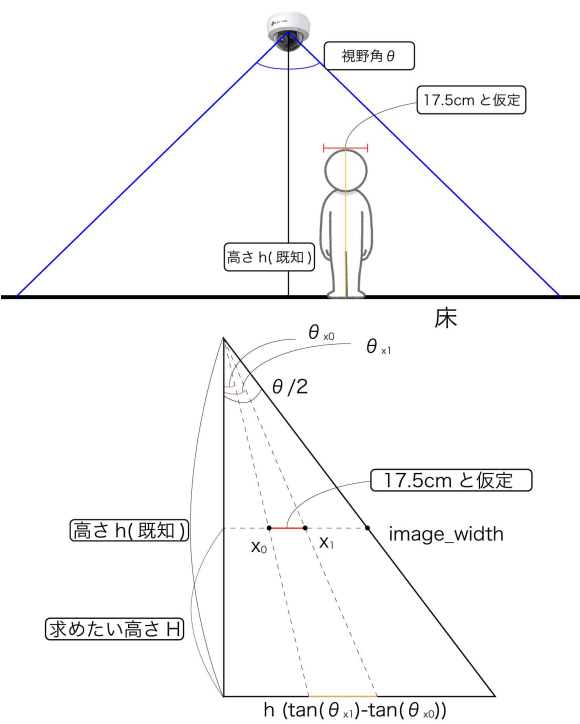


図 3: アプローチ 2 の高さ推定



図 4: Unity 上でのアバタ表示結果 (左) と現実空間の様子 (右)

を実現した。検出された頭部位置は JSON 形式でブロードキャスト送信する。Unity 上で websocket-sharp ライブラリにより WebSocket 通信を実装し、左手系座標への変換、ObjectPool パターンによる動的アバター管理、線形補間による位置更新の平滑化を実施している。図 4 に Unity 上でのアバタ表示結果を示す。

3.6 並列処理とパフォーマンス最適化

複数カメラからの映像を効率的に処理するため、マルチスレッド処理、GPU 活用 (CUDA 対応 YOLOv5s) による最適化を実施した。2 台のカメラ使用時において、サーバ側の平均処理時間は 0.28 秒に抑えられ、リアルタイム処理の要求を満たすことができた。

4. 動作確認と評価

本研究で構築した頭部位置反映システムが MR キャンパスの実現において十分な性能を有しているかを検証するため、位置精度 (現実と VR 空間の誤差)、検出精度 (頭部検出率)、システム遅延 (反映時間)、処理性能 (サーバ処理時

間) の 4 項目について定量的評価を実施した。

本システムは混雑感や臨場感の伝達を主目的とするため、厳密な位置精度よりもリアルタイム性を重視する。Konieczka らの関連研究 [4] を参考に、検出精度 85% 以上、遅延 2 秒以内、位置ずれ 17.5cm 以内 (頭部アバターのサイズと同等) を目標値とした。

4.1 実験環境

評価実験は、立命館大学大阪いばらきキャンパス (OIC) の研究室内で実施した。使用機器は Intel Core i7-11700、32GB RAM、GeForce RTX 3070 を搭載した PC (Windows11) と、TPLink VIGI C240 カメラ 2 台 (解像度 2560 × 1440、視野角 95.7° × 51.7°) である。カメラは高さ 2.64m の天井に 2.0m 離れて設置し、サーバ環境は Docker (Ubuntu 20.04) 上に構築した。

4.2 位置精度の評価

位置精度の評価では、現実世界の床面上の 1 点を原点とし、VR 空間上でも同じ位置を原点として設定した。被験者 3 名が、カメラの画角内の 4 つの座標位置 (0, 0), (−0.5, 0), (1, 0), (0, 2)[m] に移動し、その地点での現実世界の座標と VR 空間に表示されたアバターの座標の差を計測した。ここでいう誤差とは、現実世界における被験者の立ち位置と VR 空間に表示されたアバターの位置との間の xy 平面上でのユークリッド距離である。頭部の真下に足があると仮定し、例えば現実世界で (1, 0) の位置に立っている時、VR 空間のアバターが (1.05, 0.03) の位置に表示された場合、誤差は $\sqrt{(1.05 - 1)^2 + (0.03 - 0)^2} \approx 0.058[m]$ となる。

実験の結果、表 1 に示すように、各座標における位置誤差の測定で全体の平均誤差は 14.1cm であり、目標値である 17.5cm 以内を満たすことができた。

一方で、カメラの画角端に近い座標ほど誤差が大きくなる傾向が確認された。これは、カメラの歪み補正の残差や、画角端での座標変換精度の低下が要因と考えられる。

4.3 頭部検出精度の評価

頭部検出精度の評価では、カメラの画角内における頭部検出率を評価するため、8 名の被験者が 3 分間自由に移動する様子を撮影し、その動画からランダムに 30 フレームを抽出して分析を行った。各フレームにおいて、実際の頭部数と検出された頭部数を比較し、検出精度 (正しく検出された頭部数/実際の頭部数) と誤検出率 (誤検出数/検出総数) を算出した。

表 1: 各座標における VR 空間との誤差 [cm]

	(0, 0)	(-0.5, 0)	(1, 0)	(0, 2)
被験者 1	11.7	8.0	3.1	10.6
被験者 2	28.8	28.6	8.4	14.1
被験者 3	12.0	14.1	16.1	13.6
平均	17.5	16.9	9.2	12.8

30 フレームの分析結果、検出精度は 81.6%であり、目標の 85%には届かなかった。検出漏れの主要因として、頭部検出アルゴリズムが頭部のみを検出対象としている点が考えられる。人物の体全体がカメラの画角内に映っていても頭部が画角から外れたり、オクルージョンで隠れたりした場合に、検出されないケースがあった。

4.4 システム遅延の評価

システム遅延の評価は、現実世界から VR 空間への反映遅延とサーバ側処理時間の 2 つの観点から実施した。

現実世界から VR 空間への反映遅延については、ユーザーが方向転換や停止などの動作を行った際に、その動作が VR 空間上のアバターに反映されるまでの時間を計測した。スマートフォンのカメラ (60fps) を用いて、実際の動作と VR 空間上のアバターの動きを同一画角で撮影し、フレーム数の差から遅延時間を算出した。各動作パターンにつき 10 回計測を実施した結果、フレーム数の平均誤差は 85 フレームであり、遅延時間は 1.42 秒であった。これは目標値の 2 秒以内を満たしている。

また、サーバ側の処理性能を評価するため、画像データの受信から YOLOv5s による物体検出、座標変換、データ送信までの所要時間を計測した。Python の time モジュールを使用し、各カメラについて 50 フレーム分のデータを収集した。

表 2 に示すように、サーバ側の平均処理時間は約 0.28 秒であり、リアルタイム処理に十分な性能を示している。カメラ間での処理時間の差は小さく、並列処理が効果的に機能していることが確認された。

4.5 考察

本システムの評価結果から、MR キャンパスの実現に向けて実用的な性能を概ね達成できたことが確認された。位置精度 14.1cm、検出精度 81.6%、遅延時間 1.42 秒という結果は、混雑感や臨場感をリモートユーザーに伝えるという本システムの目的において、十分実用的な水準であると考えている。

システム性能の分析から得られた重要な知見として、遅延時間の内訳が挙げられる。全体遅延 1.42 秒のうち約 80% (1.14 秒) が RTSP による映像転送で発生し、サーバ処理は 0.28 秒と全体の 20%に過ぎなかった。これは、本システムのボトルネックが計算処理ではなく通信プロトコルにあり、今後の最適化の方向性を明確に示している。また、並列処理の実装によりカメラ台数増加の影響が最小限に抑えられており、システムのスケーラビリティも確保されている。

表 2: 各カメラの処理時間 [秒]

カメラ	最小値	最大値	平均値
カメラ 0	0.16	0.67	0.28
カメラ 1	0.19	0.41	0.28

5. まとめ

本研究では、MR キャンパス構想の一環として、天井設置カメラによる頭部位置推定システムを構築し、HMD 非装着者を含む現地空間の混雑状況を VR 空間に可視化することを実現した。

評価実験の結果、位置精度 14.1cm、遅延時間 1.42 秒と目標値を達成し、実用的な性能を確保した。頭部検出精度は 81.6%と目標をわずかに下回ったが、混雑感の伝達には十分な水準である。システム分析により、遅延の約 80%が映像転送に起因することが判明し、今後の最適化の方向性が明確になった。

今後は人物追跡や頭部向き推定などの機能拡張により、より自然な MR キャンパスの実現を目指す。本研究は、物理空間とデジタル空間を融合した新たなコミュニケーション基盤の創出に向けた重要な一歩となった。

参考文献

[1] Niantic Labs. The metaverse is a dystopian nightmare. let's build a better reality, 2021. Accessed: 2025-07-09.

[2] 柳澤祥太, 中村文彦, 木村朝子, 柴田史久. リアルワールドメタバース実現に向けたクロスバース・プラットフォームの構築 (1) —3 次元マップの生成と選択的配信機構—. 日本バーチャルリアリティ学会第 30 回大会論文集, 2025.

[3] Yang Lihe, Kang Bingyi, Huang Zilong, Xu Xiaogang, Feng Jiashi, and Zhao Hengshuang. Depth anything: Unleashing the power of large-scale unlabeled data. *2024 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2024)*, pp. 10371–10381, 2024.

[4] Konieczka Adam, Balcerek Julian, Marcin Andrzejewski, Kaczmarek Szymon, and Szczygiel Jakub. Smart real-time multi-camera people locating system. *2022 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pp. 98–103, 2022.