

XR環境におけるUWBタグを用いた位置情報に基づくUI表示とデバイス制御

大谷 芳之¹

概要：本研究では、UWBタグを用いた位置情報に基づくUI表示とデバイス制御を実現するXRシステム「POISE」を提案する。POISEは、Apple Vision Proを用いてUWBタグの位置をリアルタイムで追跡し、UIを動的に表示することで、実空間のデバイスとUIの物理位置を一致させる。これにより、XR環境内での直感的デバイス操作を可能にする。本システムの有効性を検証するため、位置推定の精度評価を行い、実空間とデジタル空間を統合したインタラクションの可能性について考察する。

1. はじめに

XRと物理空間とのシームレスな接続は、XR体験を豊かにするうえで重要な課題である。従来の光学式トラッキングシステム[1]は、複数オブジェクトのリアルタイムかつ高精度な位置推定を可能とする一方、消費電力の高さや遮蔽に弱いといった課題を抱えていた。また、複数のカメラやマーカの設置が必要となり、導入コストと環境整備の負担が大きい。

本研究では、UWB(Ultra-Wideband)タグを用いた位置情報に基づくUI表示とデバイス制御を行うXRシステム「POISE(Position-Oriented Interactive System Enhancer)」を提案する。POISEは、Apple Vision Proを用いてUWBタグの位置をリアルタイムに追跡し、UIを動的に表示することで実空間にあるデバイスの操作を直感的に行えるようにする。

従来手法と比較して、POISEは準備の容易さや遮蔽・照明環境の影響の少なさ、省電力性などの利点を有する。そのため、高精度・高頻度のトラッキングが必須ではない多くの環境において、実用的なシステムを提供できる可能性が高い。

本稿では、まずPOISEの設計と実装を詳述し、続いて評価結果と考察を示す。屋内環境での位置推定が容易となり、XRアプリケーションとの統合性が高いことから、さまざまなXRシナリオでの応用が期待される。

2. 関連研究

2.1 位置情報を用いたXR環境におけるUI適応

ARIoTは、Bluetoothビーコンを用いて近傍のIoTデバイスを特定し、画像認識技術と組み合わせてトラッキングおよびAR表示を行うシステムを提案している[2]。

また、画像認識と大規模言語モデル(LLM)を統合することで、物理空間内のオブジェクトを検出し、それに基づいてUIを自動生成するシステムも報告されている。たとえばSIGMAは、視覚モデル(Detic)を活用して環境内のオブジェクトを認識し、タスク支援のためのホログラム表示を行っている[3]。さらに、AOIはMediaPipeとLLMを組み合わせ、オブジェクトの分類情報から対応するUIを動的に生成するシステムを提示している[4]。

2.2 UWBのXR分野での応用

WiFi, Bluetooth, RFID, UWBなど、多様な無線通信技術が屋内測位に応用されてきたが、なかでもUWBは高精度とコスト効率の面から有力な室内測位技術とされている[5]。

Chanwoo Leeらは、没入型パフォーマンスアートにおける小道具のトラッキングやインタラクションをUWBで実装し、Viveトラッカーのコストや遮蔽への弱さを克服して低コストかつ広範囲の安定した追跡を実現した[7]。また、Hadaらは、UWBとHTC Viveトラッカーの比較実験を通じて、UWBがマルチパス環境下でも高い精度を保持し、PDoAとTDoAを併用した手法により動的シナリオにも対応し得ることを示唆している[8]。

¹ 株式会社マープル

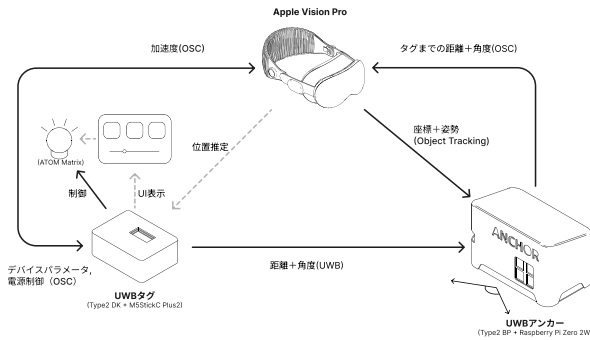


図 1 POISE システムの概要



図 2 UWB アンカーデバイス

3. システムの概要

POISE は、Apple Vision Pro を中心とした XR システムである。visionOS の ARKit による Object Tracking[6] と位置推定機能に加え、UWB アンカーとタグを組み合わせることで、位置情報の取得だけでなく、UI 表示やデバイス制御まで統合的に行う。本章では、各構成要素の詳細とアプリケーションの設計を述べる。

3.1 デバイス構成

本システムは、以下の 3 種類のデバイスを基本構成とする：

- **Apple Vision Pro**：メインデバイスとして機能し、ARKit による自己位置推定と Object Tracking を活用して UWB アンカーの姿勢・位置を高精度で追跡する。また、得られた情報を統合し、実空間と連動した UI を表示する。
- **UWB アンカーデバイス (図 2)**：村田製作所の Type 2BP (LBUA0VG2BP-EVK-P)[9] を使用し、USB 接続した Raspberry Pi Zero 2 W を介して Apple Vision Pro へデータを送信する。3D プリントした外装を Object Tracking で認識し、空間上でのアンカー位置を正確に特定する。
- **UWB タグデバイス (図 3)**：村田製作所の Type 2DK (LBUA2ZZ2DK-EVK)[10] を用い、M5Stack 社の M5StickC PLUS2[11] によりタグの移動検知と電源管理を行う。同じく M5Stack 社の ATOM Matrix[12] を制御用デバイスとして接続し、UI 操作の結果をシリアル通信で受け取り LED を制御する。1 台のアンカーで最大 5 台のタグを追跡可能。

3.2 ソフトウェア実装

3.2.1 アンカー位置の取得

visionOS の Object Tracking を利用して、Apple Vision Pro のカメラ映像に映る UWB アンカーの 3D モデルを認

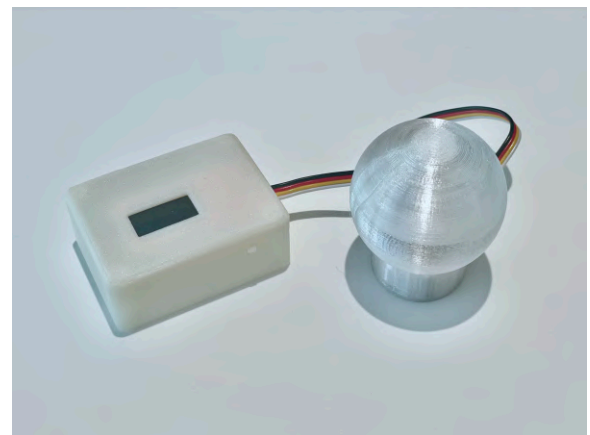


図 3 UWB タグデバイス (左) と制御用 LED デバイス (右)

識し、アンカーの座標や姿勢を取得する。ARKit と併用することで、自己位置推定を行いながら空間上でのアンカー配置を把握できる。

3.2.2 電源管理とデータ送信制御

M5StickC PLUS2 の加速度センサーによりタグの移動を検出し、移動している間のみ Type 2DK への電源供給とデータ送信を行う。4 秒以上静止すると、自動的にタグへの電源供給が停止する。

3.2.3 位置情報の取得と送信

Type 2BP から取得できる距離や方位角 (azimuth)、仰角 (elevation) などの測位データを、OSC (Open Sound Control) プロトコルを介して Apple Vision Pro へ送信する。

3.2.4 位置推定と UI 表示

受信した UWB 測位データは拡張カルマンフィルタ (EKF) によって処理され、アンカーからタグまでの相対座標が推定される。ARKit の自己位置情報と統合することで、タグの絶対座標を特定し、空間上に UI や 3D モデルを重ね合わせて表示できる。UI はユーザー視点に追従して動的に配置が変化実現する。

3.2.5 UI 操作とデバイス制御

UI 操作により、3D モデルや音源のパラメータを制御すると同時に、OSC を介して M5StickC PLUS2 へコマンド

を送信し、物理デバイスの LED を制御する。

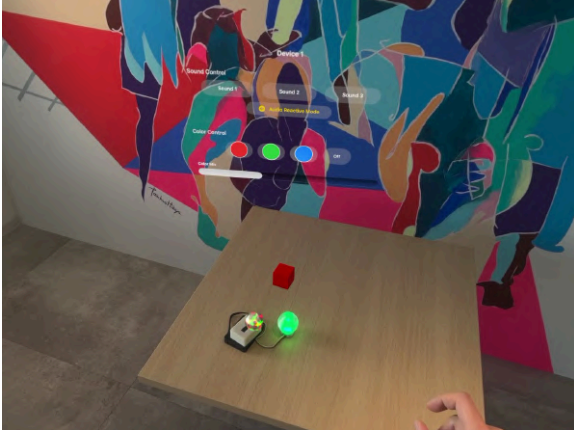


図 4 UI 表示とデバイス操作の様子

3.3 位置推定とフィルタリング手法

ARKit による自己位置推定と UWB アンカーの距離・角度測定を拡張カルマンフィルタ (EKF) で統合し、Type 2BP から取得した NLoS の判定結果と M5StickC PLUS2 の加速度センサから得られる情報を活用した適応フィルタリングを導入している。

3.3.1 座標変換と統合

UWB センサから得られた球座標系の測定値 (距離 d , 仰角 e , 方位角 a) を次式で直交座標系へ変換する：

$$\begin{aligned} x &= d \cos e \cos a \\ y &= d \sin e \\ z &= d \cos e \sin a \end{aligned} \quad (1)$$

ここで、 e は仰角 (elevation), a は方位角 (azimuth) を示す。ARKit から得られる自己位置 $\vec{p}_{\text{self}}$ と UWB アンカー位置 $\vec{p}_a$ を用いて、タグの絶対座標 $\vec{p}_{\text{tag}}$ は

$$\vec{p}_{\text{tag}} = \vec{p}_a + R(e, a) \cdot d \quad (2)$$

と表される。ただし、初期評価では高さ方向がアンカーと同一と仮定した水平面での移動を扱うため、

$$p_{\text{tag},y} = p_{a,y} \quad (\text{高さ固定モード時}) \quad (3)$$

とした。

3.3.2 拡張カルマンフィルタによる位置推定

6 次元の状態ベクトル $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ を用いた拡張カルマンフィルタ (EKF) で推定の安定性を高める。主な更新式は以下の通りである：

予測ステップ

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= F \hat{\mathbf{x}}_{k-1|k-1} \\ P_{k|k-1} &= F P_{k-1|k-1} F^T + Q \end{aligned} \quad (4)$$

更新ステップ

$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1} \quad (5)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - H \hat{\mathbf{x}}_{k|k-1})$$

ここで、状態遷移行列 F , 観測行列 H , プロセスノイズ共分散行列 Q , 測定ノイズ共分散行列 R は以下のように定義される：

- F : 等速直線運動を仮定した状態遷移を表す 6×6 行列
- H : 位置のみを観測する 3×6 の観測行列 ($[I_3 \mid 0]$)
- Q : 位置と速度のプロセスノイズを表す対角行列 (6×6)
- R : 測定値の不確かさを表す 3×3 の対角行列

3.3.3 適応フィルタリング

本システムでは、Type 2BP から取得した NLoS の判定結果に応じた測定ノイズの増減と、加速度による動的制御により、外れ値や突発的な動きに対処する。

- 測定ノイズの適応的調整：

$$R = \begin{cases} 0.05 I_3 & (\text{LoS 時}) \\ 0.50 I_3 & (\text{NLoS 時}) \end{cases} \quad (6)$$

観測ベクトルは $[x, y, z]$ の 3 次元とし、NLoS フラグをデバイス側から受け取って R の対角要素を動的に更新する。

- 加速度ベースの動的制御：加速度センサの値から重力成分を除去し、純粋な動きの加速度 $\vec{a}_{\text{real}}$ を検出する。この値の大きさに応じて速度を減衰させることで、急激な動きによる位置推定の発散を防いでいる：

$$\vec{v}_{\text{new}} = \begin{cases} 0.8 \vec{v} & \text{if } \|\vec{a}_{\text{real}}\| > 3.0 \text{ m/s}^2 \\ \vec{v} & \text{otherwise} \end{cases} \quad (7)$$

3.3.4 安定性の確保

さらに、以下の対策によってシステム全体の安定度を高めている：

- プロセスノイズの調整：位置は $Q = 0.002$, 速度は $Q = 0.001$ に設定
- 時間的整合性：更新間隔が 0.2 秒を超えた場合は、そのデータを破棄し、次の連続したデータを受信するまで前回の推定値を維持する
- 物理的制約： $\|\vec{p}\| \leq 5.0 \text{ m}$ や $\|\vec{v}\| \leq 2.0 \text{ m/s}$ などの上限制約を設ける
- 加速度データによる異常検知：重力成分を除去した後、急激変化を検出して適応的に対処

4. 評価

4.1 評価方法

アンカーと同一平面上に限定し、位置推定精度を評価した。手順は以下の通りである：

4.2 評価環境

- 評価空間：4.5 m $\times$ 4.5 m

- グリッド間隔：0.75 m ピッチ（7 × 7 グリッド）
- アンカー位置：空間の上辺を基準とし，中央（グリッド座標 [2.25, 0]）
- 測定高さ：床 0.72 m（アンカーと同一平面）

4.3 測定手順

- (1) グリッド上にある 49 点の各位置へ順次 UWB タグを設置
 - (2) タグが停止してから 4 秒待機後に位置推定データを取得
 - (3) 実際のグリッド位置の差分を記録
- この操作を 5 回繰り返し，平均値を評価指標とした。なお，計測時には毎回 Object Tracking 機能を用いてアンカーの位置を再認識し，位置合わせを行った。

4.4 評価結果

アンカーの仕様上，下方 $\pm 60^\circ$ を超える範囲では測位ができないため，以下の 2 パターンで結果を示す。

4.4.1 有効測定範囲内での性能

アンカー下方 $\pm 60^\circ$ 以内での測定結果：

- 測位精度：
 - － 平均誤差：0.31 m
 - － 標準偏差：0.15 m
 - － 67%信頼区間 (1σ)：0.36 m
 - － 95%信頼区間 (2σ)：0.59 m
- 測定の信頼性：
 - － 繰り返し精度 (1σ)：0.18 m
 - － 最小誤差：0.05 m
 - － 最大誤差：0.74 m

4.4.2 全測定範囲での性能

評価空間全体における結果：

- 測位精度：
 - － 平均誤差：0.37 m
 - － 標準偏差：0.23 m
 - － 67%信頼区間 (1σ)：0.39 m
 - － 95%信頼区間 (2σ)：0.76 m
- 測定の信頼性：
 - － 繰り返し精度 (1σ)：0.17 m
 - － 最大繰り返し精度 (1σ)：0.47 m

有効測定範囲 ($\pm 60^\circ$) 内での平均誤差は 0.31m であり，UI 表示用途として実用的な精度を達成している。95%の測定点が誤差 0.587 m 以内に収まり，繰り返し精度も平均 0.17m と安定した結果が得られた。

しかし，ARKit による自己位置推定のドリフトや Object Tracking の初期位置合わせ誤差，マルチパス環境の影響により，一部で観測値のばらつきが生じていると考えられ，今後の改善課題として挙げられる。

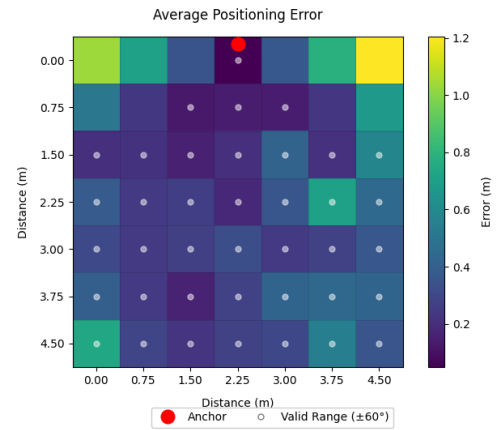


図 5 位置推定誤差の空間分布（ヒートマップ）

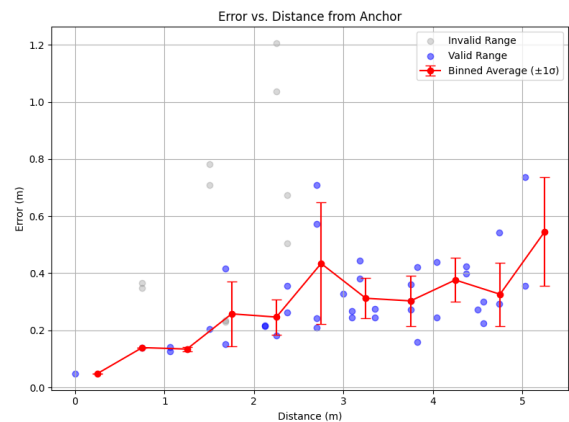


図 6 アンカーからの距離と推定誤差の関係

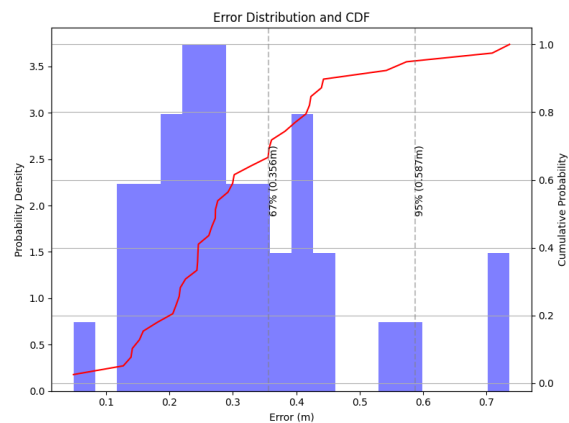


図 7 誤差分布と累積確率

5. 考察

本研究では，visionOS の ARKit による Object Tracking と自己位置推定を UWB 測位と組み合わせることで，光学式トラッキングシステムほど複雑なキャリブレーションや専用環境を必要とせず位置推定が可能であることを示した。評価実験では有効測定範囲内において平均誤差 0.311 m

を達成し、UI 表示用途として実用的な精度を確認できている。また、光学式トラッカーと比較して、UWB は遮蔽に対する耐性や省電力性の面で優位性を持つことも確認できた。

その一方で、以下の課題が明らかになった：

- **位置推定精度の向上**：マルチパスや外れ値の影響を低減するフィルタリング手法の改善や独自の NLoS 判定手法の開発が求められる。加速度センサとの統合によってリアルタイム性の向上も期待でき、複数アンカーの導入により広域かつ安定した位置推定の実現も可能となる。
- **適応的な UI 配置の実現**：ARKit の平面検出機能を活用し、操作しやすい位置に UI を動的に移動させることで、デバイスや壁面によるオクルージョンを回避する適応的な UI が実現可能となる。これにより、実空間の文脈や操作性を考慮した最適な UI 配置が期待できる。
- **システムの省電力化と拡張性向上**：低消費電力である UWB の利点を十分に引き出せておらず、OSC を使用している現状ではシステム全体の消費電力が増大してしまう。BLE を併用した通信方式の最適化や、空間認識の高度化による測位頻度の適応的制御を行うことで、より長時間の安定運用や大規模環境への対応が期待できる。

今後は上記の課題を解決しつつ、画像認識などとの連携を深めることで、より広範な利用シーンに適応可能な XR インタラクションを目指す。特に ARKit などの空間認識機能と UI の動的な配置最適化を組み合わせることで、実空間の環境に合わせた適応的かつ直感的な操作が可能となると考えられる。

参考文献

- [1] Vive tracker (3.0): Vive united states. <https://www.vive.com/us/accessory/tracker3/>, (2024.12.20, accessed).
- [2] Dongsik Jo, Gerard Jounghyun Kim: *ARIoT: scalable augmented reality framework for interacting with Internet of Things appliances everywhere*, IEEE Transactions on Consumer Electronics, vol. 62, no. 3, pp. 334-340, (2016.10).
- [3] D. Bohus, S. Andrist, N. Saw, A. Paradiso, I. Chakraborty and M. Rad: *SIGMA: An Open-Source Interactive System for Mixed-Reality Task Assistance Research – Extended Abstract*, 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Orlando, FL, USA, 2024, pp. 889-890, (2024.05).
- [4] Mustafa Doga Dogan, Eric J Gonzalez, Karan Ahuja, Ruofei Du, Andrea Colaço, Johnny Lee, Mar Gonzalez-Franco, and David Kim: *Augmented Object Intelligence with XR-Objects.*, In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), New York, NY, USA, Article 19, pp. 1-15, (2024.10).
- [5] Fazeelat Mazhar, Muhammad Gufran Khan, and Benny Sällberg: *"Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations"*, Wireless Personal Communications, vol. 97, no. 3, pp. 4467-4491, (2017.12).
- [6] Apple Inc., Exploring object tracking with ARKit, https://developer.apple.com/documentation/visionos/exploring_object_tracking_with_arkit, (2024.12.20, accessed).
- [7] Chanwoo Lee, Kyubeom Shim, Sanggyo Seo, Gwonu Ryu, Yongsoon Choi: *"Never Tell the Trick: Covert Interactive Mixed Reality System for Immersive Storytelling"*, 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Orlando, FL, USA, 2024, pp. 1009-1010, (2024.03).
- [8] Hada, Ryotaro and Arun, Aditya and Bharadia, Dinesh and Sakuta, Misaki and Saruwatari, Shunsuke: *"Demo: UWB localization and Tracking for XR Applications"*, MOBISYS '24: 22nd Annual International Conference on Mobile Systems, Applications and Services, pp. 604-605, (2024.06).
- [9] Murata Manufacturing Co., Type 2DK — NXP based UWB modules , <https://www.murata.com/ja-jp/products/connectivitymodule/ultra-wide-band/nxp/type2bp>, (2024.12.20, accessed).
- [10] Murata Manufacturing Co., Type 2DK — NXP based UWB modules, <https://www.murata.com/ja-jp/products/connectivitymodule/ultra-wide-band/nxp/type2dk>, (2024.12.20, accessed).
- [11] M5STACK, M5StickC PLUS2 ESP32 Mini IoT Development Kit, <https://shop.m5stack.com/products/m5stickc-plus2-esp32-mini-iot-development-kit>, (2024-12-20, accessed).
- [12] M5STACK, ATOM Matrix ESP32 Development Kit, <https://shop.m5stack.com/products/atom-matrix-esp32-development-kit>, (2024-12-20, accessed).