



## 自己位置推定結果の補完手法

M2 田川幸汰

### 1 自己位置推定結果の補完

実際のカメラ入力を用いたリアルタイム動作では、特徴点マッチング処理による遅延により推定位置が端末の実際の位置と乖離する可能性がある。さらに、屋内環境においては常に十分な特徴量が得られるとは限らず、マッチングが不安定となり自己位置が更新されない場合も想定される。このような課題に対して、より安定した自己位置推定を実現し、実用的な AR 道案内システムを構築するため、本研究ではセンサ情報との併用を検討した。具体的には、Apple が提供する AR フレームワークである ARKit が有する Visual Inertial Odometry (VIO) を併用し、推定結果の補完を図る。

#### 1.1 Visual Inertial Odometry

ARKit では、カメラ画像に含まれる視覚情報と、デバイスの慣性センサーによるモーション情報を統合した Visual-Inertial Odometry (VIO) によって、デバイスの位置および姿勢を推定する。これにより、特徴点マッチングのみに依存した手法に比べて、特徴点が乏しい環境や動きが速い状況においても安定したトラッキングが可能となる。

#### 1.2 ARKit のカメラ変換行列

ARKit では、各フレームのデバイスの姿勢は、カメラ変換行列 (`ARFrame.camera.transform`) で表される。この変換行列はカメラ座標系から、ARKit 座標系の変換を表す  $4 \times 4$  の同次変換行列であり、次のように書ける：

$$\mathbf{T}_{\text{cam} \rightarrow \text{AR}} = \begin{bmatrix} \mathbf{R}_{\text{cam} \rightarrow \text{AR}} & \mathbf{t}_{\text{cam} \rightarrow \text{AR}} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (1)$$

ここで、 $\mathbf{R}_{\text{cam} \rightarrow \text{AR}}$  はカメラ座標系から ARKit 座標系への回転行列、 $\mathbf{t}_{\text{cam} \rightarrow \text{AR}}$  は ARKit 座標系におけるカメラ原点位置を表す並進ベクトルである。

ARKit 座標系は、ARKit のセッション開始時に呼び出される座標系で、初期状態では重力と反対方向を Y 軸、カメラの向いている方向を Z 軸とし、Y 軸と Z 軸の外積が X 軸として固定される。ここでユーザーが定義する世界座標系を含めた各座標系の関係を図 1 に示す。

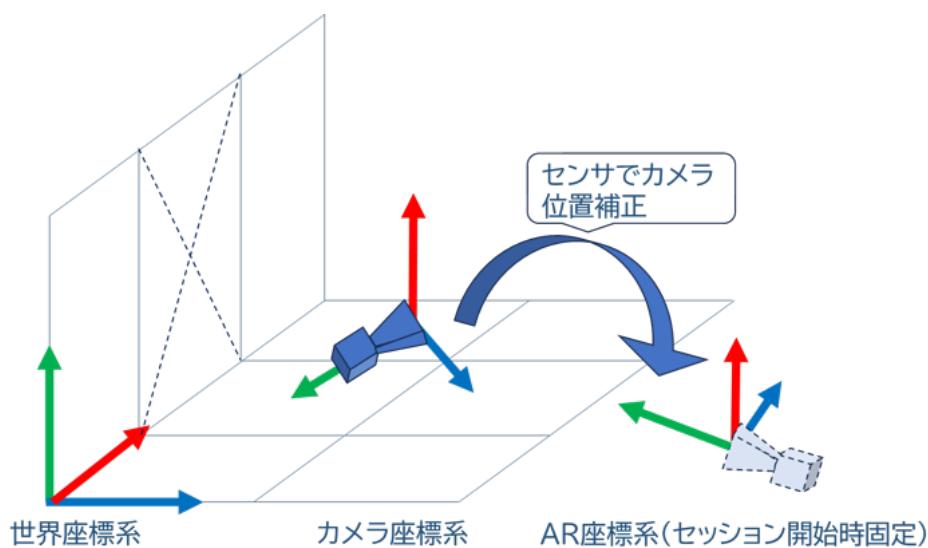


図 1: 座標系の関係



ここで、注意が必要なのは、カメラ座標系の軸の符号が異なることである。OpenCV のカメラ座標系が X 右、Y 下、Z 奥が正の向きであるのに対して、ARKit のカメラ座標は X 右、Y 上、Z 手前が正の向きとなる（図 2）

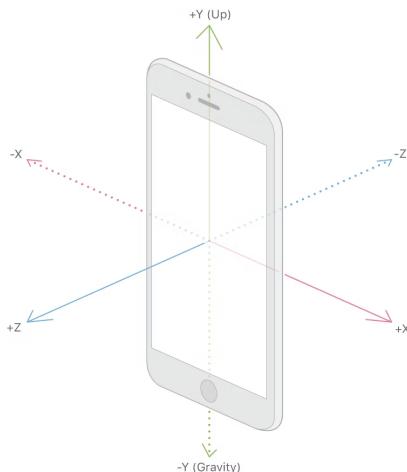


図 2: ARKit カメラ座標の定義

ここで示した座標系の違いを考慮して、自己位置推定結果の補正を行う必要がある。

### 1.3 補完の方針

カメラ画像に基づく推定結果はフレーム毎に得られるわけではなく、時々更新されるのみである。一方 ARKit による自己位置推定は高頻度で得られるため、ARKit によるカメラ位置姿勢により、適宜自己位置を更新する。本手法における補完処理の方針を以下に示す。

1. 画像取得時点の `ARFrame.camera.transform` より、カメラ座標系 → ARKit 座標系の変換行列 ( $\mathbf{R}_{\text{cam} \rightarrow \text{AR}}, \mathbf{t}_{\text{cam} \rightarrow \text{AR}}$ ) を取得する。
2. 画像の特徴点マッチングにより推定された、世界座標系 → カメラ座標系への変換行列 ( $\mathbf{R}_{\text{world} \rightarrow \text{cam}}, \mathbf{t}_{\text{world} \rightarrow \text{cam}}$ ) を取得する。
3. 上記 1, 2 の結果より、ARKit 座標系 → 世界座標系の変換行列 ( $\mathbf{R}_{\text{ar} \rightarrow \text{world}}, \mathbf{t}_{\text{ar} \rightarrow \text{world}}$ ) を計算する。ARKit 座標系と世界座標系との対応関係は、カメラ位置が更新されても不变である。
4. フレーム更新時：ARKit が毎フレーム提供するカメラ姿勢 ( $\mathbf{t}_{\text{cam} \rightarrow \text{AR}}$ ) を取得し、3 で求めた変換行列を適用することで世界座標系におけるカメラ位置を推定する。

### 1.4 変換行列の計算

ARKit 座標系 → 世界座標系の変換行列 ( $\mathbf{R}_{\text{ar} \rightarrow \text{world}}, \mathbf{t}_{\text{ar} \rightarrow \text{world}}$ ) の計算方法を以下に示す。

カメラ座標系 → ARKit 座標系の変換は以下の式 2 で表される。

$$\mathbf{X}_{\text{ar}} = \mathbf{R}_{\text{cam}1 \rightarrow \text{ar}} \mathbf{X}_{\text{cam}1} + \mathbf{t}_{\text{cam}1 \rightarrow \text{ar}} \quad (2)$$

また、世界座標系 → カメラ座標系の変換は以下の式 3 で表される。

$$\mathbf{X}_{\text{cam}2} = \mathbf{R}_{\text{world} \rightarrow \text{cam}2} \mathbf{X}_{\text{world}} + \mathbf{t}_{\text{world} \rightarrow \text{cam}2} \quad (3)$$



さらに、ARKit のカメラ座標系  $\mathbf{X}_{\text{cam}1}$  と世界座標系のカメラ座標系  $\mathbf{X}_{\text{cam}2}$  の座標系の違いは以下の式 5 で表される

$$\mathbf{X}_{\text{cam}2} = \mathbf{S} \mathbf{X}_{\text{cam}1} \quad (4)$$

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (5)$$

式 2、式 3、式 5 から、世界座標系 → ARKit 座標系の変換は以下の式 10 で整理される。

$$\mathbf{X}_{\text{ar}} = \mathbf{R}_{\text{cam} \rightarrow \text{ar}} (\mathbf{S}(\mathbf{R}_{\text{world} \rightarrow \text{cam}2} \mathbf{X}_{\text{world}} + \mathbf{t}_{\text{world} \rightarrow \text{cam}2})) + \mathbf{t}_{\text{cam} \rightarrow \text{ar}} \quad (6)$$

$$\mathbf{X}_{\text{ar}} = (\mathbf{R}_{\text{cam} \rightarrow \text{ar}} \mathbf{S} \mathbf{R}_{\text{world} \rightarrow \text{cam}2}) \mathbf{X}_{\text{world}} + \mathbf{R}_{\text{cam} \rightarrow \text{ar}} \mathbf{S} \mathbf{t}_{\text{world} \rightarrow \text{cam}2} + \mathbf{t}_{\text{cam} \rightarrow \text{ar}} \quad (7)$$

$$\mathbf{X}_{\text{ar}} = \mathbf{R}_{\text{world} \rightarrow \text{ar}} \mathbf{X}_{\text{world}} + \mathbf{t}_{\text{world} \rightarrow \text{ar}} \quad (8)$$

$$\mathbf{R}_{\text{world} \rightarrow \text{ar}} = \mathbf{R}_{\text{cam} \rightarrow \text{ar}} \mathbf{S} \mathbf{R}_{\text{world} \rightarrow \text{cam}2} \quad (9)$$

$$\mathbf{t}_{\text{world} \rightarrow \text{ar}} = \mathbf{R}_{\text{cam} \rightarrow \text{ar}} \mathbf{S} \mathbf{t}_{\text{world} \rightarrow \text{cam}2} + \mathbf{t}_{\text{cam} \rightarrow \text{ar}} \quad (10)$$

式 10 より、世界座標系と ARKit 座標系の座標変換は式 16 で表される。

$$\mathbf{X}_{\text{world}} = \mathbf{R}_{\text{world} \rightarrow \text{ar}}^{\top} \mathbf{X}_{\text{ar}} - \mathbf{R}_{\text{world} \rightarrow \text{ar}}^{\top} \mathbf{t}_{\text{world} \rightarrow \text{ar}} \quad (11)$$

$$\mathbf{X}_{\text{world}} = \mathbf{R}_{\text{ar} \rightarrow \text{world}} \mathbf{X}_{\text{ar}} + \mathbf{t}_{\text{ar} \rightarrow \text{world}} \quad (12)$$

$$\mathbf{R}_{\text{ar} \rightarrow \text{world}} = \mathbf{R}_{\text{world} \rightarrow \text{ar}}^{\top} \quad (13)$$

$$= (\mathbf{R}_{\text{cam} \rightarrow \text{ar}} \mathbf{S} \mathbf{R}_{\text{world} \rightarrow \text{cam}2})^{\top} \quad (14)$$

$$\mathbf{t}_{\text{ar} \rightarrow \text{world}} = -\mathbf{R}_{\text{world} \rightarrow \text{ar}}^{\top} \mathbf{t}_{\text{world} \rightarrow \text{ar}} \quad (15)$$

$$= -\mathbf{R}_{\text{ar} \rightarrow \text{world}} (\mathbf{R}_{\text{cam} \rightarrow \text{ar}} \mathbf{S} \mathbf{t}_{\text{world} \rightarrow \text{cam}2} + \mathbf{t}_{\text{cam} \rightarrow \text{ar}}) \quad (16)$$

## 1.5 実行結果

上記の理論に基づいてカメラ位置を更新した結果を図 3 に示す。

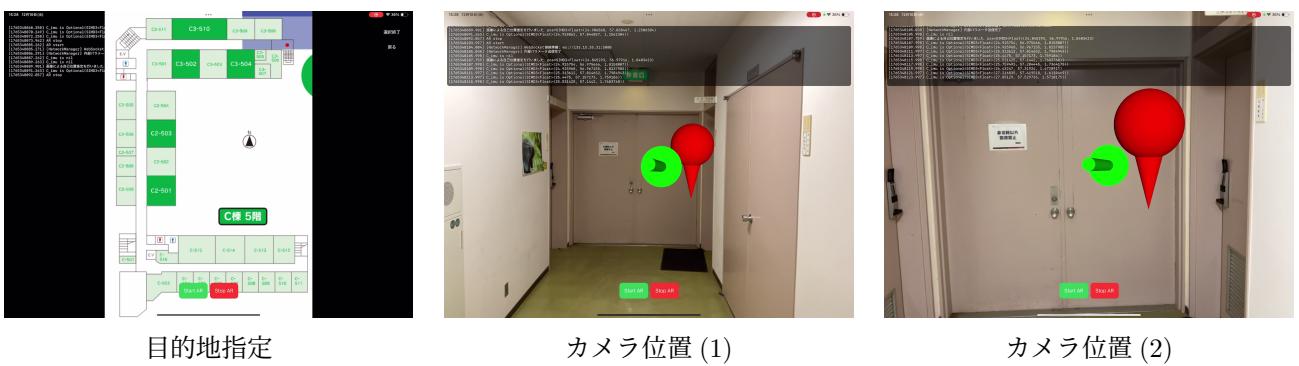


図 3: 目的地に対するオブジェクト描画結果

ユーザの世界座標は、東向きが  $X$  正、北向きが  $Y$  正、高さ方向が  $Z$  正となるように定義されている。ここで、 $C_{imu}$  は ARKit 座標系におけるカメラ位置を、ユーザの世界座標系へ変換したものである。図 3 より、ユーザが目的地方向（東向き）へ移動した際に、 $C_{imu}$  の  $X$  成分が  $25.53 \rightarrow 27.85$  と正方向に変化していることが確認できる。また、 $Y$  および  $Z$  成分についても、想定どおり更新されていることが確認された。