

ワイヤースケルトンと全方位画像による簡易モデルを用いた屋内環境での自己位置推定

M2 田川幸汰

1 はじめに

1.1 研究背景

デジタルツイン市場は急速に拡大しており、現実空間とデジタル空間をリアルタイムに連携させるためには高精度な自己位置推定技術が不可欠である。屋内で用いられる主な手法として、SLAM は LiDAR やカメラを用いて高精度な位置推定と地図生成が可能であるが、高性能なハードウェアを要する。一方、無線測位は追加機材を設置することで屋内測位が可能となるものの、運用負担が大きく導入障壁となっている。

1.2 研究目的

本研究の目的は、簡易的な三次元モデルを用いた屋内環境における自己位置推定の有効性を示すことである。全方位画像から取得したテクスチャをワイヤースケルトンに貼り付けて簡易モデルを構築し、カメラ画像との特徴点マッチングにより自己位置を推定する。加えて、その精度と実行時間が実用的であるかを、道案内システムの実装を通して検証する。

1.3 本論文の構成

本論文では、2 章で簡易なテクスチャ付き 3 次元モデルの生成手法を示す。3 章では簡易モデルと入力画像のマッチング手法、4 章ではマッチングによって得られた対応点を用いた自己位置推定手法を示す。5 章では、簡易モデルを用いた自己位置推定の実験結果について示す。さらに 6 章では、自己位置推定結果を活用した道案内システムの実装と、その精度の実用性を検証する。

2 簡易 3 次元モデルの作成

本節では、簡易 3 次元モデルを生成する手法について説明する。

2.1 ワイヤースケルトンモデルの生成

メッシュの頂点情報を入力とし、テクスチャを割り当てていない状態のモデルを生成する方法を説明する。

2.2 全方位画像からの透視投影画像生成

全方位画像座標から透視投影画像座標に変換する方法を説明する。

2.3 透視投影画像を用いたカメラ位置姿勢推定

複数方向の透視投影画像を用いた外部パラメータ推定方法を説明する。



2.4 テクスチャ取得

テクスチャ取得の方法を以下に示す。

1. メッシュ頂点の世界座標を、カメラの外部パラメータを用いてカメラ座標に変換
2. カメラ中心とメッシュ重心の距離と角度を計算し、条件を満たすメッシュだけをテクスチャの候補に
3. カメラをメッシュ重心へ向けるように調整し、透視投影画像の座標に変換
4. テクスチャの形状を、元のメッシュ形状に射影変換

3 入力画像とテクスチャの特徴点マッチング

本節では、特徴点の検出およびマッチング手法について説明する。

3.1 特徴点検出

SIFT、AKAZE、SuperPoint を用いた特徴点検出方法を説明する。SuperPoint は、画像から高品質な特徴点と記述子を同時に抽出できる学習ベースの特徴点検出器である。

3.2 特徴点マッチング

最近傍探索や SuperGlue を用いた特徴点マッチング方法を説明する。SuperGlue は、特徴点間の文脈情報を考慮して高精度に対応付けを行う学習ベースのマッチング手法である。

3.3 誤マッチ除去

Lowe's Ratio Test や RANSAC を用いて、誤マッチを除去する方法を説明する。

4 自己位置推定

本節では、自己位置推定手法について説明する。

4.1 テクスチャ画像座標から世界座標への変換

テクスチャの UV 座標とメッシュ頂点の世界座標から基底ベクトルを算出し、基底ベクトルの係数を揃えることで、テクスチャ上の 2 次元特徴点を 3 次元座標に変換する。

4.2 自己位置推定アルゴリズム

自己位置推定の方法を以下に示す。

1. カメラ画像の 2 次元特徴点とテクスチャの 3 次元特徴点を用い、直行射影誤差の大域最適化により自己位置を推定
2. 幾何学的制約に基づき、天地反転した解を除去
3. 目的関数が最小となる解を最終的な自己位置推定結果として採用

5 実験

本節では、簡易モデルを用いた自己位置推定実験に用いた機材やパラメータ、各種入力、および実験結果について説明する。

5.1 特徴点の計測と全方位画像撮影

C 棟 5 階における特徴点を計測し、4m 間隔で全方位画像を撮影した。

5.2 全方位カメラパラメータ推定結果

外部パラメータの推定結果を実測値と比較したところ、カメラ位置の誤差はおおむね 10cm 以下であることを確認した。

5.3 モデル生成結果

壁面や扉付近など特徴量が乏しい領域には、ポスターを貼付して特徴量を追加したモデルを作成し、その効果を検証した。

5.4 特徴点マッチング結果

学習ベース手法は、一般的な手法と比較して特徴が乏しい画像でも比較的安定してマッチングできた。一方、入力画像が複数のテクスチャ画像と類似した特徴を含む場合、どちらの手法でも正しいマッチングが困難であった。

5.5 自己位置推定結果

動画の各フレームを抽出し、特徴点マッチングと自己位置推定を実施した。初回は全テクスチャとマッチングを行い、2 回目以降は直前の自己位置から近傍のテクスチャに限定してマッチングを行った。学習ベース手法では、特徴点マッチングが行われたフレーム数が一般手法の約 5 倍となり、ほぼすべてのフレームで自己位置を推定できた。1 フレームあたりの計算時間は約 0.3 秒であり、特徴点マッチングに時間の大部分がかかっていることが確認できた。

5.6 目的地までの矢印描画

フロアマップから目的地を選択し、推定した自己位置から目的地までの矢印を入力動画に描画した。特徴点マッチングに成功した場合、AR 描画も正しく反映されることを確認した。

6 自己位置推定結果を用いた AR 道案内システム

実際のカメラ入力を用いてリアルタイムに動作させる場合、特徴点マッチング処理に起因する遅延により、推定位置が端末の実際の位置とずれる可能性がある。また、屋内環境では常に十分な特徴量が得られるとは限らず、マッチングに失敗して自己位置が更新されない状況も想定される。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。

6.1 AR 道案内システムの構成

本節では、構築した AR 道案内システムの構成について説明する。

6.1.1 目的地設定

画面上にマップ画像を表示し、ユーザーはタップ操作により目的地および経路上の右左折ポイントを選択できる。選択したポイントは、ホモグラフィ行列を用いて画像座標から世界座標に変換する。

6.1.2 端末と PC 間の通信

WebSocket を用いて端末と PC を接続し、カメラフレーム、内部パラメータ、センサ情報に基づき更新された自己位置を PC へ送信する。また、PC 側で計算された自己位置推定結果を非同期に受信し、目的地の AR 表示および自己位置の更新に利用する。

6.1.3 センサによる自己位置の更新と補正

Apple が提供する AR 開発フレームワークである ARKit は、センサ情報と画面上の特徴点の追跡を組み合わせ、毎フレーム端末の位置と姿勢を推定する。端末側では、この位置姿勢変化をもとに、PC から受信した自己位置情報と統合することで自己位置を更新する仕組みを構築している。現在、座標変換および統合処理は実装中である。

6.1.4 AR オブジェクト描画処理

目的地を示すピンオブジェクトと、現在位置から目的地方向を指し示す矢印オブジェクトを AR 空間に描画する。

6.2 動作結果と評価

本システムは現在実装途中であり、センサ情報に基づく自己位置更新処理は未完成である。センサのみを用いた位置推定ではドリフトが蓄積するため、PC から正確な自己位置推定結果が得られたタイミングで姿勢・位置をリセットする仕組みを導入することが望ましい。

7 まとめ

本節では、本研究の成果と今後の展望について述べる。

7.1 本研究の成果

本研究では、簡易モデルを用いた屋内環境での自己位置推定手法を提案した。提案手法を用いた道案内 AR アプリケーションの実装を通じて、自己位置推定結果が実用的な精度で取得できることを確認した。

7.2 今後の展望

今後は、AR 道案内システムを完成させるとともに、Visual SLAM など既存の自己位置推定手法と比較することで作成コストや精度を評価したい。