

カメラ画像と姿勢情報を利用したスマートフォン向け位置推定システムの研究 A Study about Location System using Orientation and Camera Image for Smartphone

打越 大成[†] 岩本 健嗣[†] 松本 三千人[†]
Uchikoshi Daisei Takeshi Iwamoto Michito Matsumoto

1. はじめに

近年、スマートフォンが急速に普及している。平成 24 年度の情報通信白書[1]によると、スマートフォンは 2009 年頃から本格的に普及しはじめ、フィーチャーフォンを含む高機能携帯電話端末市場において、販売台数比率は 2011 年には約 27%に達しており、2015 年には 50%を超え、更に拡大していく見通しとなっている。そして、スマートフォンは携帯性が高く、多彩なセンサを搭載しているため、位置情報を利用したアプリケーション[2]が多く製作されている。

現在、スマートフォンで利用されている主な位置推定手法は、GPS による位置測位、Wi-Fi アクセスポイント等の無線基地局からの電波強度を利用した位置測位、加速度センサ等を用いた自律測位に大別できる。GPS による位置測位は、4 つ以上の GPS 衛星からの電波を利用する。そのため、電波遮蔽物の多い場所や屋内では、電波を受信できなくなったり、マルチパスが発生し、測位精度が悪化したり、測位できなくなる欠点がある。無線基地局からの電波強度を利用した位置測位は、GPS を用いた位置測位に比べ、測位までの時間が短い長所があるが、測位粒度を細かくすることが難しい。加速度センサ等を用いた自律測位は、応答性が高いが、誤差が蓄積される欠点がある。

一方、人間は、屋内であっても視覚情報から誤差の蓄積なく位置を把握できる。例えば、見えているモノの相対的な位置関係や遠近感などから自身の位置を把握している。そこで、本稿では、既存の位置推定手法の問題点を解決するために、カメラから得られる画像を用いた位置推定に着目した。しかし、画像による位置推定は計算コストが高く、スマートフォンでの実時間処理が難しい。そこで、多くのスマートフォンに搭載されている、加速度、地磁気センサから得られる姿勢情報を用いることで、画像処理による位置推定の計算コストを低減する手法を提案する。

2. 提案手法

一般的に、画像による位置推定は、画像から SIFT[3]や SURF[4]等によって検出された特徴点を予め蓄積された特徴量と実空間上の座標が対応したデータベースから検索し、特徴点のカメラ画像上の座標とデータベースに登録された実空間上の座標から、幾何学的に推定する。

本研究では、姿勢情報を用いる 2 つの高速化手法を提案する。1 つ目は、姿勢情報を用い、位置推定計算を簡略化することで、特徴点検出数を減らし計算コストを下げる。2 つ目は、データベース検索時に姿勢情報を使うことで計算コストの低減と精度の向上を図る。

まず、特徴点の各座標の関係と位置計算簡略化について

[†] 富山県立大学 大学院 情報システム工学専攻
Toyama Prefectural University Graduate School of
Engineering Information Systems Engineering

述べる。カメラの姿勢情報である方位、ピッチ、ロールを r_1, r_2, r_3 とし、カメラの実空間上の 3 次元座標を t_1, t_2, t_3 としたとき、ある点 p の実空間上の 3 次元座標 X, Y, Z とカメラを基準とした 3 次元座標 X', Y', Z' は式 1, 2, 3 を用いて、式 4 の関係がある。

$$\begin{aligned} s_1 &= \sin r_1 & s_2 &= \sin r_2 & s_3 &= \sin r_3 \\ c_1 &= \cos r_1 & c_2 &= \cos r_2 & c_3 &= \cos r_3 \end{aligned} \quad (1)$$

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$= \begin{pmatrix} c_1 c_3 + s_1 s_2 s_3 & s_1 c_2 & -c_1 s_3 + s_1 s_2 c_3 & t_1 \\ -s_1 c_3 + c_1 s_2 s_3 & c_1 c_2 & s_1 s_3 + c_1 s_2 c_3 & t_2 \\ c_2 s_3 & -s_2 & c_2 c_3 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = M \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (4)$$

また、カメラ画像上の 2 次元座標 x, y とカメラを基準とした 3 次元座標 X', Y', Z' は、カメラの焦点距離 F とカメラ画像の縦横サイズ w, h を用いて、式 5 の関係がある。

$$X' = \left(x - \frac{w}{2}\right) \frac{1}{F} Z' \quad Y' = \left(y - \frac{h}{2}\right) \frac{1}{F} Z' \quad (5)$$

これらの式を用いて、 $m_{11} \sim m_{34}$ の 12 の未知数を求めることで位置を推定することができる。12 の未知数を求めるには、既知の特徴点を 6 点以上検出しなければならない。そこで、本研究では、姿勢情報を与えることで、 $m_{n1} \sim m_{n3}$ を求める。これにより、未知数を 3 に減らし、2 点以上の既知の特徴点の検出による位置推定を可能とする。

次に、姿勢情報を使うことによるデータベース検索の高速化と精度向上について述べる。カメラには、視野角があるため、カメラの方向がわかれば、写りえる点を絞り込むことができる。よって、本研究では、スマートフォンの姿勢情報によって、カメラの方向を決定し、写りえない点を検索データから除外する。例えば、図 1 のように、北を向いている時、点 p, q のような北向きの壁に存在する点はカメラには写りえないため、特徴量比較を行う前に除外する。特徴量同士の比較に比べ、姿勢情報の比較は高速に行える。そのため、本手法により、比較処理を高速化できる。また、本来写りえない点を除外するため、誤って見えない点と対応させることを防ぐことができ、精度を向上できる。

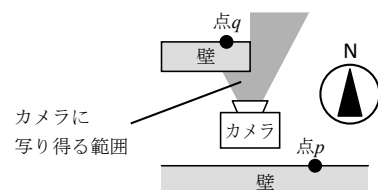


図 1 カメラに写り得ない点のイメージ

3. 評価用システムの実装

本稿では、位置推定システムを Android アプリケーションとして実装した。特徴点検出、比較部には OpenCV[5]による ORB[6]を用いた。

既存の特徴点検出では、特徴量が視認方向によって変化する。そのため、同じ場所にある点であっても、視認方向が異なれば、異なった点として収集する必要がある。しかし、視認方向の多少の変化であれば、同じ点として認識できる。また、姿勢情報による検索データの絞り込みを各点情報毎に行っては効率が悪い。そのため、本実装では姿勢情報をいくつかの状態に離散化し、時勢状態毎にまとめて管理する。

姿勢状態を多く設定するとより多くのデータを除外できる。しかし、より多くの姿勢状態で特徴点を収集する必要がある。また、姿勢状態を少なく設定すると、データの絞り込みの割合が少なくなり効果が薄れる。また、现阶段では、人間が特徴点を収集し、位置情報を付加するため、人間が理解しやすい姿勢状態が好ましい。そのため、実装したシステムでは、姿勢状態を重力方向、方位に対して軸対象な 14 状態に離散化した。14 状態は、カメラのレンズ方向が、実空間上に定義した座標系の各軸上に頂点がある正八面体とその双対の関係にある正八面の角頂点方向を向いた姿勢とし、最も近い姿勢に振り分けた。しかし、特徴点収集時の姿勢状態と位置推定時の姿勢状態が近くても、しきい値を跨いでいた場合、異なる姿勢として認識され、検索されなくなる。これを防ぐために、実装したシステムでは、位置推定時は、状態の近い 3 姿勢のデータから比較を行う。また、本システムでは、最大 65535 データを扱えるように実装した。

4. 評価

実装したシステムでは、姿勢情報を用いる事によって比較する特徴量のデータ数を 3/14 に削減している。そのため、理論値で 4.67 倍に高速化できる。しかし、実際には姿勢情報からデータを絞り込む処理やカメラ画像を読み込む処理がオーバーヘッドとなるため、理論値ほどの高速化はできない。本実験では、提案手法による高速化の有効性を確認するために、姿勢情報を利用しない既存の画像による位置推定システムを用意し、実装したシステムと比較した。

実験には、SO-01C、F-04E を使い、位置推定を 100 回行った時の平均処理時間を比較した。オーバーヘッドの比率は比較用データ数、特徴点の検出数によって変化するため、32700 点のデータと 65534 点のデータを用意し、特徴点検出数は 25, 50, 100 で行った。ただし、データの内容は処理時間には影響がないため、本実験で用意したデータはランダムで生成したデータである。

実験結果を表 1, 2 に示す。SO-01C, 比較データ数 65534, 特徴点検出数 100 の時、平均高速化率 3.37 倍で最も高速化でき、F-01E, 比較データ数 32100, 特徴点検出数 25 の時、平均高速化率 1.25 倍で最も高速化の比率が低くなった。理論的な最大高速化率 4.67 倍には届かなかったが、いずれの端末、比較データ数、特徴点検出数で、処理時間を短くすることができ、提案手法の有効性が確認できた。

5. おわりに

本稿では、姿勢情報を利用した画像によるスマートフォン向け位置推定手法を提案した。そして、実験では提案手法の有効性が確認できた。

今後、提案手法での位置推定精度を確認する必要がある。また、提案手法では、より少ない特徴点検出数で位置推定を行える可能性がある。そのため、適切な特徴点検出数を求めることができれば、更に高速化できる。他にも、加速度センサ等を用いた自律測位との組み合わせも考えられる。また、人間が特徴点を収集し、位置情報を付加する作業は、時間がかかり、現実的ではない。そのため、三次元復元技術等を用いて、自動化する必要がある。

表 1 SO-01C での提案手法と既存手法の処理時間の比較

比較データ数	特徴点検出数	提案手法		既存手法		平均高速化率
		平均処理時間 [msec]	標準偏差 [msec]	平均処理時間 [msec]	標準偏差 [msec]	
32700	25	432	40	662	37	1.53
	50	519	16	1075	29	2.07
	100	655	39	1781	42	2.72
65534	25	279	41	1056	31	2.20
	50	611	42	1737	57	2.84
	100	931	39	3136	53	3.37

表 2 F-01E での提案手法と既存手法の処理時間の比較

比較データ数	特徴点検出数	提案手法		既存手法		平均高速化率
		平均処理時間 [msec]	標準偏差 [msec]	平均処理時間 [msec]	標準偏差 [msec]	
32700	25	272	43	339	46	1.25
	50	294	49	453	14	1.54
	100	314	22	723	86	2.31
65534	25	261	17	453	28	1.73
	50	326	24	690	150	2.12
	100	408	30	997	168	2.44

参考文献

- [1] 総務省, “スマートフォン等の急速な普及と端末市場の変化”, 情報通信白書, <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h24/html/nc122110.html> (2013).
- [2] Foursquare, <https://ja.foursquare.com/>.
- [3] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features”, International Conference on Computer Vision, Vol.60, No.2, pp. 91-110 (1999).
- [4] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, “Speeded-Up Robust Features (SURF)”, Computer Vision Image Understanding, Vol.110, No.3, pp.246-259 (2008).
- [5] OpenCV, <http://opencv.org/>.
- [6] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, “ORB: an efficient alternative to SIFT or SURF”, In Proc. International Conference on Computer Vision (2011).
- [7] 出口 光一郎, “射影幾何学による PnP カメラ補正問題の統一的理解”, 情報シンポジウム, Vol.90, pp.41-50 (1990)
- [8] 大江 統子, 佐藤 智和, 横矢 直和, “画像特徴点によるランドマークデータベースを用いた動画からのカメラ位置・姿勢推定”, 電子情報通信学会技術研究報告.PRMU, パターン認識・メディア理解, Vol.104, No.572, pp.19-24 (2005).