

# 特徴量マッチングを用いた屋内 AR ナビゲーションシステムの実装

2005MT003 青山 輪太郎      2005MT054 小嶋 亮吾

指導教員 河野 浩之

## 1 はじめに

近年、携帯電話などの小型端末の処理能力が向上し、それらに向けた様々なアプリケーションが開発されている。その中でも拡張現実 (以下, AR) を用いたナビゲーションシステムが注目されている。AR ナビゲーションシステムとは、端末に搭載されているカメラの映像を使用し、矢印や、イメージタグの情報をカメラの映像上に表示することでリアルタイムで目的地までのナビゲーションをするシステムである。従来の AR ナビゲーションシステムでは、GPS や電子コンパスなどのセンサーによって位置情報を取得していたが、そのどちらのセンサーも屋内での正確な情報を取得することは難しく、屋内で正確なナビゲーションを行うことが出来なかった。表 1 は既存の AR インターフェースを比較したものである。ほとんどのものは屋内でのナビゲーションを想定したものではなく、屋内でのナビゲーションを考慮している junaio もマーカーを用意しなければならない。

本研究の目標は、Wi-Fi 機器の登録情報から GPS が使えない場所でも位置情報を取得することが出来る PlaceEngine<sup>\*1</sup> によって取得した荒い位置情報をもとに、カメラから得られた画像に対して画像処理を行うことで正確な位置情報を推定し、その結果をもとに屋内でも高精度なナビゲーションができるシステムを実現することである。実現するにあたって、Harlan Hile ら [1] の提案した目印となるオブジェクトの特徴量を基準とする手法を取ると共に、動画像として特徴量を追跡することによって位置情報の誤差を修正する。実装のテストとして、今回は南山大学瀬戸キャンパス内にて各教室間のナビゲーションを行うことにした。

表 1 既存の AR ブラウザの比較

ブラウザ名	対応端末	特徴	位置認識制度
セカイカメラ	Android iPhone	各ユーザーが自由に付加した エタグを表示	GPS の電波強度 やセンサーの精度 により変化 (数メートル程度の誤差)
wikitude	Android iPhone	Wikipedia や Qype に収録されている 位置情報を利用	
Wikitude Drive	Android	カーナビに AR ブラウザを使用することで 低コストを実現 (まだ安全性に問題点有り)	
Metro Paris Subway	iPhone	駅の乗り換えや駅周辺の店の情報を表示	
Layar	Android	表示させるコンテンツの種類を選択して表示可能	
junaio	Android iPhone	画像認識 (LLA マーカー) を使い 屋内でも情報を表示可能	

<sup>\*1</sup> <http://www.placeengine.com/>

## 2 AR インターフェースの現状

### 2.1 表示誤差の修正

AR インターフェースを用いたナビゲーションアプリは GPS と電子磁気コンパスによって位置情報を取得しているが、その位置情報には少なからず誤差が存在する。特に、屋内での正確な位置情報の取得は困難であり、精密なナビゲーションを行う際にはそれらの誤差を修正する必要がある。図 1 はナビゲーション表示に生じた誤差の例である。本来ならばナビゲーションカーソルは通路に沿って表示されるべきだが、図 1 の場合は階段に向かうため通路を無視して壁に向かっていている。

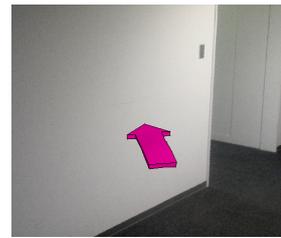


図 1 ナビゲーション表示の誤差

### 2.2 GPS やセンサーから生じる誤差

従来の AR インターフェースナビゲーションシステムでは、GPS と電子磁気コンパスのみを用いて位置情報を取得しているため、屋内でのナビゲーションが困難であった。PlaceEngine のように無線 LAN からの情報を用いて位置情報を取得するアプリケーションも存在するが、そのみで正確なナビゲーションを行える精度の位置情報を取得することは出来ない。また、仮に屋内で GPS の情報を正確に取得できたとしても、一般的に GPS の誤差はおおよそ 6m 程であり、屋内では通路の幅以上の誤差となり得るため、ナビゲーションする場合は致命的である。

## 3 画像処理を用いた屋内ナビゲーションシステムの提案

屋内における正確な位置情報の取得が困難であることから、従来の AR インターフェースナビゲーションシステムでは正確な屋内のナビゲーションは難しかった。本研究では、GPS による位置情報の取得が困難な屋内にて正確な AR インターフェースナビゲーションシステム

の開発を目指す。

### 3.1 屋内での位置情報の取得

GPS では、地図上の広範囲の位置情報を調べるのに適しているが、屋内の位置情報を得ることは難しい。また、複数階建ての建物での階層を知ることも出来ない。そこで本研究では、PlaceEngine と屋内に設置された Wi-Fi 機器を用いることによってリアルタイムに現在位置を測定することにした。

#### 3.1.1 PlaceEngine を用いた位置情報の取得

PlaceEngine で、周囲にある無線 LAN アクセスポイントから MAC アドレス及び電波強度情報を得て、各アクセスポイントからの電波強度のバランスをもとに現在地を算出する。図 2 は PlaceEngine の概念図である。図 2 の示す流れを説明すると次のようになる。

##### [step1]

PlaceEngine クライアントが付近にある Wi-Fi 機器から MAC アドレスや電波の強度を取得する。

##### [step2]

端末から step1 で取得した Wi-Fi 機器の MAC アドレスおよび電波強度をネットワークを通じてインターネット上に存在する PlaceEngine Server に送信する。

##### [step3]

PlaceEngine Server から先に送信した Wi-Fi 機器の情報から割り出した位置情報を受け取る。

PlaceEngine はインターネット上にある PlaceEngineServer に登録されている無線 LAN 機器の MAC アドレス等を基準に位置情報の取得を行う。そのため、位置情報を取得するためにはあらかじめナビゲーションを行う領域内の Wi-Fi 機器の情報を PlaceEngine Server に登録しておかなければならない。Wi-Fi 機器の登録は、登録したい Wi-Fi 機器を通じて“PlaceEngine クライアント”を使用すれば良い。

### 3.2 特徴量マッチングによる位置情報の推定

本研究では、カメラから取得した画像を使用することによって位置情報やカメラ姿勢を推定する。

#### 3.2.1 局所特徴量の抽出

局所特徴量は、ピクセル間のベクトルを取得し、それらの関連を特徴量とする手法で、スケーリングや回転などの変化に大して強固であるという特徴を持っている。局所特徴量には、SIFT, PCA-SIFT, SURF などがある。

図 3 は、画像から SURF 特徴量を抽出したものである。抽出された特徴量は、一つ一つが独立した特徴量ベ

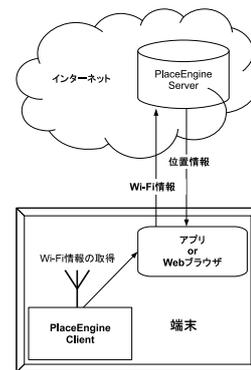


図 2 PlaceEngine の概念図

クトルを持っており、個別に識別することが可能である。



図 3 局所特徴量抽出アルゴリズムを用いた特徴点の抽出

#### 3.2.2 局所特徴量抽出手法の比較

本研究では、イメージマッチングを行うことで位置情報の推定を行うため、正しい結果を得るためには適切な特徴量を選択しなければならない。特に、カメラ映像をリアルタイムで処理する必要があるため、毎フレームごとの計算量は少ない物を選ぶ必要がある。

表 2 局所特徴量抽出手法の比較

	SIFT 特徴量	PCA-SIFT 特徴量	SURF 特徴量
抽出速度 [ms]	1036	1135	354
次元数	128 次元	36 次元	64 次元
探査速度 [ms]	314	124	91
認識精度	97.2%	98.5%	95.0%

表 2 は本道らによる“大規模画像認識のための局所特徴量の性能比較”[4]のデータと Herbert Bay による“Speeded-Up Robust Features (SURF)”[5]のデータを端的にまとめたもので、サイズが 640\*480 の画像 10000 枚を用いてイメージマッチングの手法を比較したものである。SIFT 特徴量は、典型的な局所特徴量のアルゴリズムによって得られるが、抽出速度は SURF 特徴量に、認識精度は PCA-SIFT 特徴量に、ベクトルの次元数は PCA-SIFT, SURF のどちらよりも高く、処理速度に問題がある。そのため、本研究には向いていないと判断し

た。PCA-SIFT 特徴量は認識精度がもっとも高く、次元数も 36 次元なのでデータ量、比較速度ともに良好だが、抽出速度が非常に低速である。そのため、リアルタイムで特徴量を抽出する必要がある本研究では不向きである。SURF 特徴量は、認識精度は PCA-SIFT 特徴量に劣るが、それでも十分な認識率を示しており、また抽出速度は SIFT 特徴量、PCA-SIFT 特徴量と比べて最も高速である。比較速度、次元数についても、3 章 2 節 4 項で述べる手法を用いることで十分にカバーすることが出来る。よって、本研究では SURF 特徴量を用いてイメージマッチングを行う。

### 3.2.3 画像情報データベースと特徴量データベースの構築

イメージマッチングによって位置情報を推定するため、画像から抽出された特徴量と位置情報を関連付けたデータベースを構築する。データベースを構築するに当たって、あらかじめ撮影された画像ファイルと関連する位置情報を用意し、SURF 特徴量抽出アルゴリズムを用いてあらかじめ特徴量を抽出しておく。それらのデータを作成するために、本研究では USB カメラを使用して撮影とデータ入力を同時に行うことが出来るソフトの開発を行った。そのソフトを用いて、位置情報データベースと特徴量データベースを構築する。

SURF 特徴量は 64 次元のベクトルから構成されており、ベクトルを比較することで類似する特徴を探査する。多数の特徴に対して単純な線形探索を行うと、その次元数の高さから計算量が膨大になってしまうので、本研究では、多次元ベクトルの比較に適している黄瀬ら考案した“局所特徴量のハッシングに基づく大規模画像検索”[3]の手法、を SURF 特徴量に用いてデータベース構築することによってマッチングの高速化を図る。具体的な手順は次のとおりである。

64 次元からなる SURF の特徴ベクトルを  $x = (x_1, x_2, \dots, x_{64})$  とし、 $\mu_i$  はデータベースに収められた  $i$  次元目の値の平均値とする。このとき、

$$u_i = \begin{cases} 1 & \text{if } x_i - \mu_i \geq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

式 (1) で定められるビットベクトル  $u$  を考える。このビットベクトルを用いて、 $x$  のハッシュ関数  $h(x)$  を式 (2) のように定める。

$$h(x) = \left( \sum_{i=1}^d u_i 2^{(i-1)} \right) \bmod H_{size} \quad (2)$$

ここで得られたハッシュ値を元に、チェイン法によってハッシュテーブルを作成する。なお、特徴量を現すビットベクトルとその特徴量を含んでいた画像の ID を

要素を持った特徴量構造体を作成することで元画像と特徴量をリンクしている。

## 4 画像マッチングと PlaceEngine を組み込んだナビゲーションシステムの構造

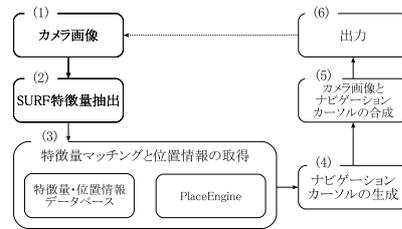


図 4 システム全体図

システム全体の構成は図 4 のようになる。以下が図 4 の解説である。

- (1) カメラから画像を取得する。
- (2) (1) で取得した画像から特徴量を抽出する。
- (3) (2) で抽出した特長量をデータベース内の特徴量と比較し、位置情報を得る。その際、PlaceEngine を用いることにより大まかな位置情報を取得し、データベースを探索する範囲を絞る。
- (4) (3) で得た位置情報を元にナビゲーションカーソルを生成する。
- (5) (4) で生成したナビゲーションカーソルをカメラ画像と合成する。
- (6) (5) で合成された画像を映像として出力し、(1) に戻る。これら一連の処理をフレーム毎に行う。

また、位置情報を推定する (3) の仕組みは図 5 のようなものになる。

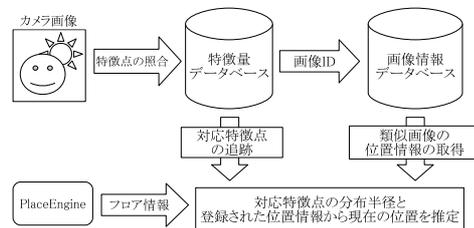


図 5 位置情報推定の流れ

図 5 は、まず、カメラからの入力画像から特徴量を抽出し、その特徴量を特徴量データベースと照合し、類似画像を探し出す。探し出された類似画像の ID をもとに画像情報データベース内の位置情報を取得し、その位置

情報から実際の位置情報を推定する。また、特徴量データベースで類似画像を探索する際に PlaceEngine を用いることによってデータベースを分割することで、類似画像の探索精度を向上させている。

## 5 性能評価

実験には、南山大学瀬戸キャンパスの G 棟および H 棟の画像データベースを作成し、うまく位置情報が推定できるかを検証した。画像情報データベースに用いる画像は解像度が 640\*480 のカラー画像 720 枚と、それらを大津の手法で 2 値化したものを用いた。PC は CPU が Intel(R) Core(TM)2 Duo 2.5Ghz, メモリが 2.99GB のものを使用した。位置情報の取得精度の検証は、南山大学瀬戸キャンパス G 棟の 3 階と H 棟の 3 階の 70 箇所の地点から位置情報の推定を試み、その地点での成功率から検証を行った。位置情報の推定精度として、観測地点から位置情報の推定結果が安定するまで待機し、その推定結果が正しい位置を示していれば成功、正しくなければ失敗と見なし、各環境における成功率を精度とした。

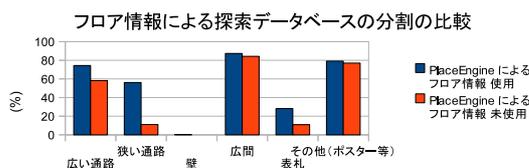


図 6 PlaceEngine の有無による性能の比較

図 6 は PlaceEngine の有無による性能の評価を行ったものである。PlaceEngine を用いた場合は、推定精度が高い順に「広間」87.4%、「その他(ポスター等)」79.1%、「広い通路」74.4%、「狭い通路」56.1%、「表札」28.2%、「壁」0% であった。対して PlaceEngine を用いなかった場合は「広間」84.2%、「その他(ポスター等)」77.0%、「広い通路」58.2%、「狭い通路」11.1%、「表札」11.0%、「壁」0% で、PlaceEngine を用いた場合位置情報の推定精度が向上することが分かった。

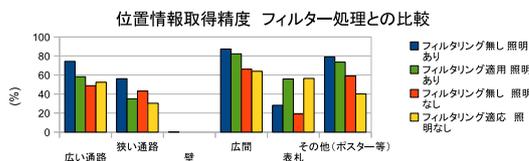


図 7 2 値化フィルタの有無による照明による影響に対する性能の比較

図 7 は照明による風景の変化に対する強度を 2 値化フィルタを適用した場合と適用していない場合を比較

したものである。「表札」を除くほとんどの場合 2 値化フィルタを適用しない場合の方が位置情報の推定精度は高い。ただし、照明の有無による位置情報の推定精度の変化量は 2 値化フィルタを適用した場合は少なくなるため、照明による風景の変化に対する強度は 2 値化フィルタを適用した場合のほうが高いことが分かった。

## 6 まとめ

本研究では SURF 特徴量を用いた特徴量マッチングと PlaceEngine を用いて位置情報を推定することが可能なシステムを開発し、いくつかの環境で 2 値化フィルタ処理を適用した場合など、様々な状況において性能評価を行った。その結果、理想的な環境では最大で 87.4% の精度で位置情報を取得することができた。また、フィルタ処理を画像に適用することによって、ある程度の照明に対する風景の変化に対する強度が向上することが分かった。しかし、類似風景によって起こる問題、Wi-Fi 機器の不足によるフロア情報の取得によって起こる問題、手ブレによる特徴点の取得の失敗によって起こる問題など多くの解決できなかった課題が残り、この方式のみによる屋内のナビゲーションは困難であると感じた。今後は、これらの課題を克服するために、ジャイロセンサーによる補助を用いた精度の向上や、PlaceEngine で利用することができる Wi-Fi 機器の追加などを行うことによって問題点を解決したい。

## 参考文献

- [1] Harlan Hile, Radek Grzeszczuk, Alan Liu, Ramakrishna Vedantham, Jana Kosecka and Gaetano Borriello: "Landmark-Based Pedestrian Navigation with Enhanced Spatial Reasoning," Lecture Notes in Computer Science, Vol.5538, pp.59-76, 2009.
- [2] 都築勇司, 藤吉弘巨, 金出武雄, "SIFT 特徴量に基づく Mean-Shift 探索による特徴点追跡," 情報処理学会 研究報告, No.157, pp.101-108, 2007.
- [3] 黄瀬浩一, 岩村雅一, 中居友弘, 野口和人, "局所特徴量のハッシングに基づく大規模画像検索," DBSJ journal, No.8, pp.119-124, 2009.
- [4] 本道貴行, 黄瀬浩一, "大規模画像認識のための局所特徴量の性能比較," 画像の認識・理解シンポジウム 論文集, No.5, pp.550, 2008.
- [5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "Speeded-Up Robust Features," Computer Vision and Image Understanding, Vol.110, pp.346-359, 2008.