

## 位置情報付き路上画像の撮影方向推定システムの提案

丸山 拓馬<sup>†1</sup> 柳井 啓司<sup>†1</sup>

GPS 位置や撮影時の方向を利用するアプリケーションや web サービスが多く出現している。しかし GPS 端末は普及してきたものの方向情報を記録するデバイスはまだまだ少ない。そこで本研究では撮影時の方向情報を推定するシステムを実装した。提案システムはストリートシーンを対象として、GPS 位置情報が既知の画像を路上パノラマ画像と比較して撮影方向を推定する。方向推定には局所画像特徴の SIFT マッチングに基づく手法を利用した。マッチングペアの検出には、近似最近傍探索法の Best-Bin-First(BBF) アルゴリズムを用いた。またマッチングペアのノイズ除去方法について、3 種類のフィルタを用意してその結果を考察している。本研究はリアルタイムに方向推定が利用可能である web 上で利用可能なシステムの実装を目指しており、そのプロトタイプを実装した。

### The System for Estimating Directions of Geotagged Street Photos

TAKUMA MARUYAMA<sup>†1</sup> and KEIJI YANAI<sup>†1</sup>

Recently, there are many applications and web services which require not only GPS location information but also camera direction information of photos. However, few devices to record camera directions of photos. Then, we make a system which estimates camera directions of given photos which have GPS location information. To estimate directions, the proposed system employs SIFT based matching. We apply the Best-Bin-First method to search for the nearest matching pairs. Removing some irrelevantly-matched pairs, we prepare three types of filters. The filters improve accuracy on estimation of camera direction. We implemented a prototype system which can estimate camera directions of given geotagged photos as a Web-based interactive system.

<sup>†1</sup> 電気通信大学 情報理工学研究所 総合情報学専攻

Department of Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications

## 1. はじめに

### 1.1 背景

画像、動画に位置情報をつけて利用するシステムが増えている。撮影した GPS 情報付きの画像を利用した地図上に旅行アルバムを表示するシステムや、カメラを通してランドマークの位置・方向を見る iPhone の SekaiCamera<sup>\*1</sup> や GoogleAndroid の GoogleGoggles<sup>\*2</sup> などがそれに当たる。拡張現実感<sup>\*3</sup> とよばれる現実空間にコンピュータを用いて情報を付加提示する技術でも、撮影時の位置方向情報の利用や特定物体をマークアップを利用する。これらシステムや技術を利用する際に、GPS 位置や方向を検出するセンサや位置方向をマークアップするフレームワークが必要となる。しかし、GPS は十分普及してきているが GPS だけでは撮影位置は分かっても撮影方向は知ることができない。また撮影方向を知ることは地磁気センサーによって可能であるが、GPS 付きカメラに地磁気センサーを搭載したデバイスはほとんど存在しない。そこで、本研究では位置情報付き画像に方向情報を付加するための研究を行った。

## 2. 目的

本研究では、GPS 位置情報付きの画像を入力を受け、その位置での方向推定を行う。またリアルタイムに計算を行う web ベースのシステムを実装する (図 1)。入力には位置情報付き画像、出力は推定方向の表示である。つまりアップロードされた画像の位置情報を解読して、自動的に方向推定のための計算を行い、計算された方向を GoogleMap と GoogleStreetView で表示する。

方向推定手法には画像特徴によるマッチングを利用した。マッチングの対象とする位置情報付きの路上画像には、GoogleStreetView<sup>\*4</sup> の路上パノラマ画像を利用する。

\*1 SekaiCamera

<http://sekaicamera.com/>

\*2 GoogleGoggles

<http://google.com/mobile/goggles/>

\*3 Augmented Reality(AR)

\*4 <http://www.google.co.jp/help/maps/streetview/>



図 1 実装したシステム  
Fig. 1 web based system

### 3. 関連研究

#### 3.1 パノラマ画像と画像認識

パノラマ画像に対する位置・方向の推定には、既存研究がいくつか存在する。近年の方向推定手法では、SIFT 特徴<sup>1)</sup> を利用した上野らの研究<sup>2)</sup> がある。この論文では、ユーザの現在位置・方向情報の精度改善を目的として、街並み画像データベースのマッチング手法を提案している。建物の幾何拘束 (エピポーラ拘束) を考慮してマッチング点の選択を行うことで、従来の最近傍探索手法より精度の高い改善を示している。小野らの研究<sup>3)</sup> では、撮影地点が既知となる参照画像群を用いた入力画像の撮影位置同定を行っている。画像分割した

ブロック特徴と SURF との比較を行い、投票数や認識率に関して興味深い考察がなされている。

また、ユーザが動画を用いた際のリアルタイムな位置合わせには、興相らの研究<sup>4)</sup> がある。この研究では興相らが撮影したパノラマ画像群をデータセットに用いている。特徴的なのがリアルタイムでの動画内の要素へのタグ付けを行っているという点である。システムはリアルタイム性を実現させるため、近隣でのパノラマとのマッチングを並列して行うベストパノラマアノテーション手法を提案している。またマッチングには輝度の平均二乗による判定を施している。

#### 3.2 路上画像に関して

路上で撮影画像を扱う際に、効率的な物体検出を行う研究はすでになされている。その一例として、Derek Hoiem らの研究<sup>5)</sup> が挙げられる。Derek らは路上の歩行者や車などの物体検出に、物体検出器によらない検出位置推定のフレームワークを提案した。道路、空、建物の領域と表面形状を三分割して、さらに建物の領域をいくつかの部分領域に指定することで歩行者や車が検出される確率の高い場所を推定している。また 3D モデルとして物体認識を行い、歩行者のスケールの推定にまで言及できる特徴的な手法である。

### 4. 提案システムの概要

撮影した GPS 位置情報付き路上画像をアップロードすることにより、視線方向を推定して、結果を GoogleMap と GoogleStreetView 上に表示するシステムについて説明する。このシステムの処理の流れは以下のようになる。

#### (1) 画像のアップロード

- 路上で撮影した GPS 位置情報付きの JPEG 画像をアップロードする。

#### (2) パノラマ画像の取得、統合

- GoogleStreetView より撮影位置に隣接する画像を取得。部分画像を統合してリサイズする。

#### (3) 視線方向の推定

- アップロードされた画像、パノラマ画像から SIFT 特徴を抽出。比較してマッチングさせる。

#### (4) 結果の表示

- ブラウザ上で Google Map に撮影した位置とその撮影方向を表示する。また Map 表示の隣に近接 GPS 地点で見える撮影画像の方向を表示する。

表 1 exif ヘッダの GPS 情報  
Table 1 part of exif gps header

タグ	タグ名	意味
0	GPSVersionID	タグのバージョン
1	GPSLatitudeRef	緯度の南北
2	GPSLatitude	緯度 (度、分、秒)
3	GPSLongitudeRef	経度の東西
4	GPSLongitude	経度 (度、分、秒)
-	-	-
16	GPSTimeZone	撮影方向 (yaw)

## 5. 提案システムの詳細

### 5.1 画像のアップロード

システムの web ページから画像ファイルをアップロードする。アップロードには Exif ヘッダに GPS 情報を保持する JPEG 画像を用いる。表 1 に Exif のヘッダ形式を示した。GPSLatitude と GPSLongitude を取得することで、撮影時の位置を記録する。また、GARMIN の OREGON 550<sup>\*1</sup> や iPhone3GS など撮影時の方向を記録するデバイスで撮影した場合には GPSTimeZone タグが付加されるので、システム評価時の推定位置との誤差の評価に利用する。

### 5.2 パノラマ画像の取得、統合

パノラマ画像を Google Maps API<sup>\*2</sup> を利用して取得する。ブラウザのリクエストとしてハッシュキーを生成して、これを利用することでサーバーから各種パノラマ画像を取得して、それら統合する。図 2 に取得された画像の一例を示した。生成されたパノラマは画像特徴抽出の際に左右の両端地点の特徴が、滑らかに抽出できるように、右端にマージを持たしている。

### 5.3 視線方向の推定

以下のような 3 つの手順で視線方向の推定を行う。

- (1) パノラマ画像とクエリ画像の両方から画像の局所特徴を抽出する
- (2) 2 つの画像間で局所特徴のマッチングを行う

\*1 GARMIN OREGON 550

<https://buy.garmin.com/shop/shop.do?pID=26875>

\*2 <http://code.google.com/intl/ja/apis/maps/>



図 2 パノラマ画像の取得  
Fig. 2 Getted panorama image

- (3) マッチングした結果から、視線方向を算出する

### 5.4 画像特徴

まず方向推定を行うにはパノラマ画像とクエリ画像のどの部分が一致しているか判定しなくてはならない。よって部分的なマッチングに利用できる、画像局所パターンを記述する特徴量で記述を行う。具体的な特徴量として条件を満たすものには SIFT<sup>1)</sup> や SURF<sup>6)</sup> などが当てはまり、本研究では前者の SIFT を選択した。SIFT (Scale Invariant Feature Transformation)<sup>1)</sup> は画像中の特徴的なキーポイントとスケールを特定して、特徴点まわりの画像局所パターンを 128 次元の特徴ベクトルで表現する。スケール、回転、視点変化に不変な特徴である。

### 5.5 マッチング法

局所特徴でのマッチングの際、パノラマ画像と通常の画角 50 度程度のカメラ写真の比較と言う点から、歪みによる特徴量の誤差が考えられる。だが、実験に利用した Google Street View でのパノラマ画像 (図 2) は空に近い部分と地面に近い部分では物体の歪みがあるが、中心部分では SIFT の特徴マッチングに耐えうる程度の歪みである。よって視線の水平方向を算出する限りに置いてはそれほど問題はないと想定して、特に補正は行っていない。

マッチング方法は特徴を kd-tree で構成してその近隣点より求める Best Bin First アルゴリズムを利用する。また最後に、マッチング点のノイズペアを除去してマッチングを終了する。

#### 5.5.1 kd-tree

kd-tree とは、k 次元のユークリッド空間にあるデータ群を分類する空間分割データ構造のことである。空間を超平面で凸集合に再帰的に分割していく Binary Separate Tree の特殊ケースとみなせる。具体的な構成方法は次のようになる。まず、すべての特徴ベクトルを囲

む超長方形を考え、それに対応する根ノードを1つ設ける。次に特徴空間の1つの次元に閾値を設けて空間を2分割する。分割した各々の超長方形に含まれる特徴ベクトルが一つになった時点で終了する。最終的に得られるすべての領域はベクトルが一つのみ含まれており、これをノードとした木が完成する。これを kd-tree と呼ぶ。

### 5.5.2 BBF

Best Bin First(BBF)<sup>7)</sup> は Jeffrey S. Beis らによって提案された kd-tree に最近傍探索(NearestNeighbor:NN)を用いた近似アルゴリズムである。まず、抽出した特徴点で kd-tree を構築する。query として与えられた特徴点はこの kd-tree に従い、構造上ではなく距離的に近い bin から順に探索していく。

またこのアルゴリズムは近傍点をどのくらい検出したところで打ち切るかで、精度と速度をトレードオフにできる。本研究では近傍点が200点検出されたところで探索を打ちきるように設定した。この閾値を適用した根拠は D.Lowe の研究<sup>1)</sup> で言及されている。近隣点200点で打ちきるように実装したところ、100,000のキーポイントデータベースで実験した際に最近傍点のマッチにならないケースは5%未満だったと結論している。

### 5.6 マッチングノイズの除去

BBF アルゴリズムを用いて選出された特徴点ペアは1つの特徴量として最適なペアであっても、画像間で検出されたペアとの関係で良質なマッチング点と見なせない場合がある。例えば、同じような局所パターンが至る所に存在する場合や他の特徴点に比べ位置関係や角度、大きさが大きく異なっている場合がそれに該当する。

この問題を解消するために本研究では三つの処理を施した。

#### 5.6.1 双方向マッチフィルタ

通常のマッチングでは一方の画像から検出された特徴点群より、1つの kd-tree を構成して最近傍探索を行いマッチングペアを検出する(single match)。双方向フィルタではこの一方向的なマッチングを見直し、双方で互いに最近傍となるペアを残す。つまりパノラマ画像とクエリ画像と二つの kd-tree を構築して、両方でマッチングを行う。このフィルタにより特定のパターンが多く検出される画像(例えば煉瓦や鉄柵など)で、この様なパターンをノイズ除去できると考えられる。

#### 5.6.2 SIFT 方向マッチフィルタ

SIFT 特徴は回転に対してしてロバストな記述法であり、その検出時に特徴点の回転方向を検出する。SIFT 方向フィルタではこの回転角を利用して、ペア同士の回転差が  $\pi/3$  以上異なっているものペアを除去した。これは空が上になるように撮影されていることを前提

とするが、一般的に空が上にこない画像は少なく、本研究の実験ではそういった画像を含まないためにこのフィルタを利用した。これにより位置関係によらないミスマッチの削減を行える。

#### 5.6.3 RANSAC マッチフィルタ

パノラマ画像と撮影画像にはマッチング点位置に射影関係があるので、エピソード拘束をもちいた除去を行った。SIFT 方向フィルタとは異なる位置的关系を加味したフィルタである。具体的には Random Sampling Consensus(RANSAC) アルゴリズムを利用して、幾何的位置関係の悪いマッチ点の除去を行った。

RANSAC とはある数学モデルに沿う様に与えられたデータを選別するランダムサンプリングを利用した手法であり、本研究の RANSAC フィルタではエピソード幾何の Fundamental 行列を計算するために RANSAC を用いた。計算して得られた outlier 点を除去して、inlier 点を残すことがこのフィルタが行う作業である。

おおよそ 5,000 ~ 30,000 個検出された特徴点から、100 ~ 500 くらいのマッチングペアが求められる。双方向フィルタを利用することで 10 ~ 50 程度に抑えられる。さらに回転差 60 度未満の特徴のみ残すと 0 ~ 20 程度にまで削減できる。それぞれのフィルタを単体でかけた場合とすべて適用した場合を図 3、図 4、図 5、図 6、図 7 に示す。図 7 については、双方向フィルタと方向フィルタをかけてマッチングペアを減らしたのちに RANSAC フィルタを適用している。

### 5.7 視線方向の算出

アップロードされた画像の GPS 位置情報に対して、最も近い StreetView のパノラマ座標からの視線方向を推定する。

評価基準は検出された特徴点が多く含まれる領域ほど好ましいとする。また特徴点が多くなる領域での特徴点の平均方向を求め、この方向を視線方向の中心として算出する。

判定に利用する具体的な領域の大きさは画角 60 度とした。これは一般に普及しているデジタルカメラの画角が 50 度程度であり、領域を分割した際に扱いやすい切りの良い数値であることによる。

本手法では、まず 60 度の領域を 30 度毎にずらして、特徴点を最多に含む領域を決める。最多な領域が複数検出される場合には、その中に含まれる特徴点ペアのユークリッド距離を求め全体の和が最小の領域を選び出す。選んだ領域で特徴点の方向平均値を求めこれを推定値と定める。

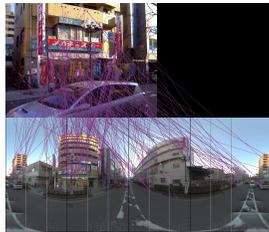


図 3 単方向マッチフィルタ  
Fig. 3 single direction match



図 4 双方向マッチフィルタ  
Fig. 4 dual direction match

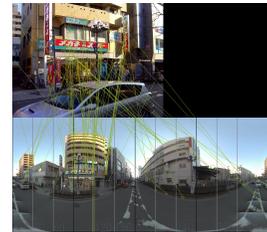


図 5 SIFT 方向マッチフィルタ  
Fig. 5 SIFT direction match



図 6 RANSAC マッチフィルタ  
Fig. 6 RANSAC match

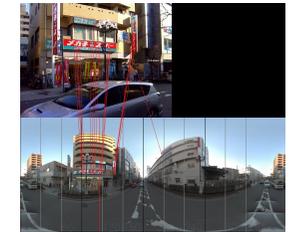


図 7 すべてのフィルタ  
Fig. 7 all\_filter match

## 5.8 結果の表示

結果はブラウザで表示する。実行した結果をパノラマ画像と比較することで方向値を算出する。得られた結果の角度は Google Map と Google Street View に表示する。表示した様子を図 1 に示す。

## 6. 実験

### 6.1 実験方法

路上で取得したパノラマ画像 451 枚の方向推定の結果を評価した。主に東京にある京王線調布駅を中心とした周辺の路上画像を撮影した。撮影には方向情報を付加することが可能な GARMIN の OREGON550 を利用した。このカメラは表 1 にある GPSTagDirection タグに情報を付加する。この方向情報を利用して、推定方向との誤差を評価に利用した。また各評価画像でマッチング点を除去しない場合を含め、双方向フィルタ、方向マッチフィルタ、RANSAC フィルタを組み合わせ計 8 パターンで実験を行った。なお RANSAC フィルタを他のフィルタと併用する際には、すべて最後に適用した。

### 6.2 実験結果

ブラウザでの実行時間は約 50 秒ほどで、そのうち SIFT 特徴抽出の計算に要する時間が 40 秒以上と大半を占める。また SIFT 計算に長い時間を費やすのは、パノラマ画像が  $3584 \times 1536$ 、入力画像が  $2048 \times 1536$  となりそのサイズの大きさに原因がある。

各種フィルタでの方向推定値誤差分布や許容角度とその正答率を図 8、図 9 に示す。正答率は双方向マッチング (dual) と方向情報マッチング (direction) を組み合わせたものが最もよく、 $30^\circ$  の許容角度で 35% の精度を達成している。

また、各種フィルタを用いた場合での特徴点の検出数については図 10 の様になった。

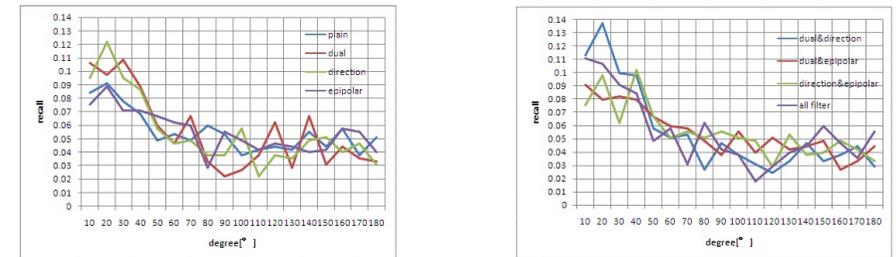


図 8 各種フィルタでの推定値誤差の分布  
Fig. 8 error distribution of the estimated direction

RANSAC マッチフィルタ、双方向マッチフィルタ、SIFT 方向マッチフィルタの順に検出数が増えていく。またフィルタを組み合わせた例では、全体のマッチ点が大幅に減少しており、すべてのフィルタを組み合わせるとほとんどの点が適合しなくなってしまう。

マッチングペアの数と許容誤差の関係は図 11 に示す。マッチング数が多い画像は許容誤差は小さくなると予想されるが、実際にはそうならなかった。各フィルタで特徴的なのはマッチングペアの検出数が一定であるものがある一方で、その分散が大きなものも存在している。例えば、すべてのフィルタを利用した例では検出数がゼロか 7 以上かで明確に分かれている。これは RANSAC アルゴリズムに利用する検出点のペア数が 8 個以上でないとは動作しないため、入力が 8 個未満の場合にマッチング点を 0 としていることが大きく起因している。一方で単体の双方向マッチフィルタや方向マッチフィルタは幅広い分布を持ち、フィルタを組み合わせるとマッチ点数はある定数付近に停滞する。

全体を俯瞰してみると図 11 の angle error の値が小さくなる領域に収まることが望まし

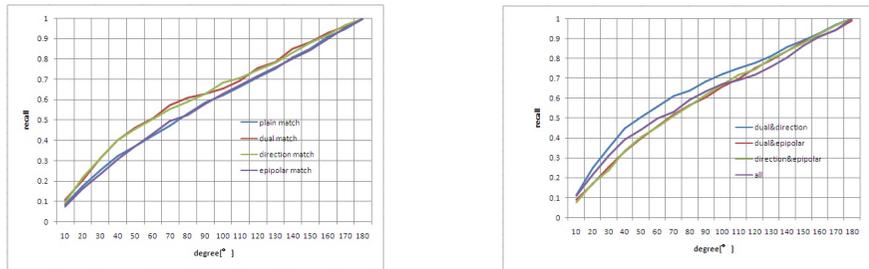


図 9 許容角度と正答率

Fig.9 correct rate and acceptance degree

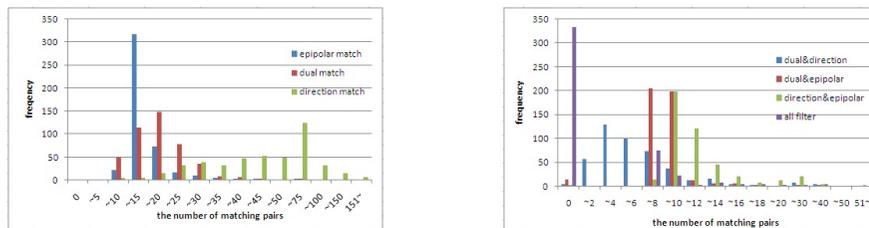


図 10 マッチング点の検出数

Fig.10 the number of matching pairs

いと考えられる。しかしこの条件を満たす分布は縦に広がりを持たず、横に広がりを持っていないてはならない。実験結果はこの条件を満たすことはできなかった。だがすべてのフィルタを適用した例について考えるなら、検出点ペア数が 0 となる場合を除いた適合率をみると質のよいマッチングを高い確率で残すことが可能になる。適合率と再現率は常にトレードオフとなるが、本実験でもその傾向が確認できた。

最後に図 21 にマッチング点の一致例をいくつか挙げる。上二段がマッチングがうまく行った例であり、その一段下がマッチングがうまく行っていない例である。また図 25 はアメリカ東海岸のニューヨークおよびフィラデルフィアで撮影された画像であり、このシステムがストリートビューの存在する世界のどの場所からも利用可能な様子を示している。

## 7. 考 察

実験では正解が完全に信用できるものと仮定していたが、実際には GoogleMap 上に表示

した時に誤差が判明するケースが多々ある。例えば GoogleMapAPI にクエリとしてリクエストする GPS 位置と streetview でのパノラマ画像位置には誤差がある。この位置のずれの分布は図 26 にしめす。これより、GPS 位置情報と StreetView の位置誤差は大体 3 割程度が 2m 以内におさまり、8 メートルまで許容するとほぼ 9 割の画像を含むことが分かる。車線の幅が 3 メートルくらいなので、これにパノラマ間の距離を 10m 程度とすれば標準的な誤差の平均は 4 ~ 5m 程度で妥当だと分かる。だがこれは撮影時の GPS にまったくずれがないと仮定しており、実際にはこれに加えて GPS 撮影位置の誤差が影響する。ビルや障害物が多い場所ではこの問題は特に大きくなる。

これに加え本実験での厳密さを阻害する要因もいくつかある。例えば正解方向として利用した方向値も機器に依存して予期せぬ方向を記録してしまうことがままたあり、数百枚の結果をブラウザで一つ一つ確認するには時間がかかり、実験データにふさわしいものかの選別は十分に行えなかった。さらに方向推定方法では画像中心からマッチ点がずれているかを見積もっていないので、最大でカメラの画角の半分程度に相当する角度がずれているケースもありえる。

厳密さの観点からは十分とはいえない推定アルゴリズムとその実験環境であったが、マッチングペアの数や誤差に関する大まかな傾向は実験結果から見て取れた。例えばノイズが多い状況下での RANSAC フィルタは有効に機能していない。しかしある程度点の質を改善するとマッチペアの位置関係を考慮でき、適合率を上昇させる。また双方向マッチでは基本的なフィルタとして精度改善に成功しているが、方向フィルタに対してマッチング点を減らしすぎることがある。万能であるような方向フィルタもカメラが大体地面に対して水平であるという制約化でしか動作しない。

このように各種特性のあるフィルタリングを組み合わせることでマッチングペアの質を改善することが確認できたが、ペアを減らす手法では再現率の向上に限界があると考えられる。そのためにマッチングペア生成の観点に立ち、特徴点の検出方法を見直す必要があると思われる。

## 8. ま と め

本研究では、路上で撮影した GPS 画像を StreetView 上のパノラマ画像と比較し撮影者の方向を推定する web システムを提案した。局所画像検出には SIFT、マッチングペア生成時の近傍探索には Best Bin First アルゴリズムを利用した。実験では 3 種類のフィルタを用いることで、方向マッチフィルタと双方向マッチフィルタを組み合わせた場合に許容角

30° で 35%の精度を達成した。また web 上で利用可能なシステムを実装してリアルタイムでの利用が可能であることを確認した。

### 9. 今後の課題

マッチング点に各種フィルタを用いることで精度の向上が確認できたが、再現率を大きく改善するには至らなかった。マッチング点の生成方法や新たな特徴など検討する必要がある。GPS の撮影位置の補正も含め、精度と速度の改善に挑戦したい。具体的には次の様な事に挑戦したい。

- SURF など高速な局所特徴を利用した計算時間の短縮化
- GPS 位置の補正
- 隣接した周囲のパノラマ画像でのマッチング
- パノラマ画像の歪み補正
- 歩行者や車など流動的な構造物からの検出点削除

### 参考文献

- 1) Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, Vol.60, No.2, pp.91-110 (2004).
- 2) 上野智史, 橋本真幸, 米山暁夫, 川田亮一: SIFT を用いた都市環境の建物画像検索によるユーザ位置・方向推定手法に関する一検討, 電子情報通信学会技術研究報告パターンメディア理解研究会, PRMU2009-111, Vol.109, No.306, pp.127-132 (2009).
- 3) 小野善太郎, 堀田政二, 馮堯楷, 内田誠一: 局所特徴量を用いた事例参照に基づく位置同定, 電子情報通信学会技術研究報告パターンメディア理解研究会, PRMU2009-212, Vol.109, No.418, pp.31-36 (2010).
- 4) 興梠正克, 藤田武志, 坂上勝彦, 村岡洋一: パノラマ画像群を位置合わせに用いたライブ画像上への注釈提示とその実時間システム, 電子情報通信学会論文誌, Vol.J84-D-2, No.10, pp.2293-2301 (2001).
- 5) Hoiem, D., Efros, A.A. and Hebert, M.: Putting Objects in Perspective, *IJCV*, pp. 3-15 (2007).
- 6) Bay, H., Tuytelaars, T. and Gool, L.V.: SURF: Speeded Up Robust Features, *ECCV* (2006).
- 7) Beis, J. and Lowe, D.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, *CVPR*, pp.1000-1006 (1997).

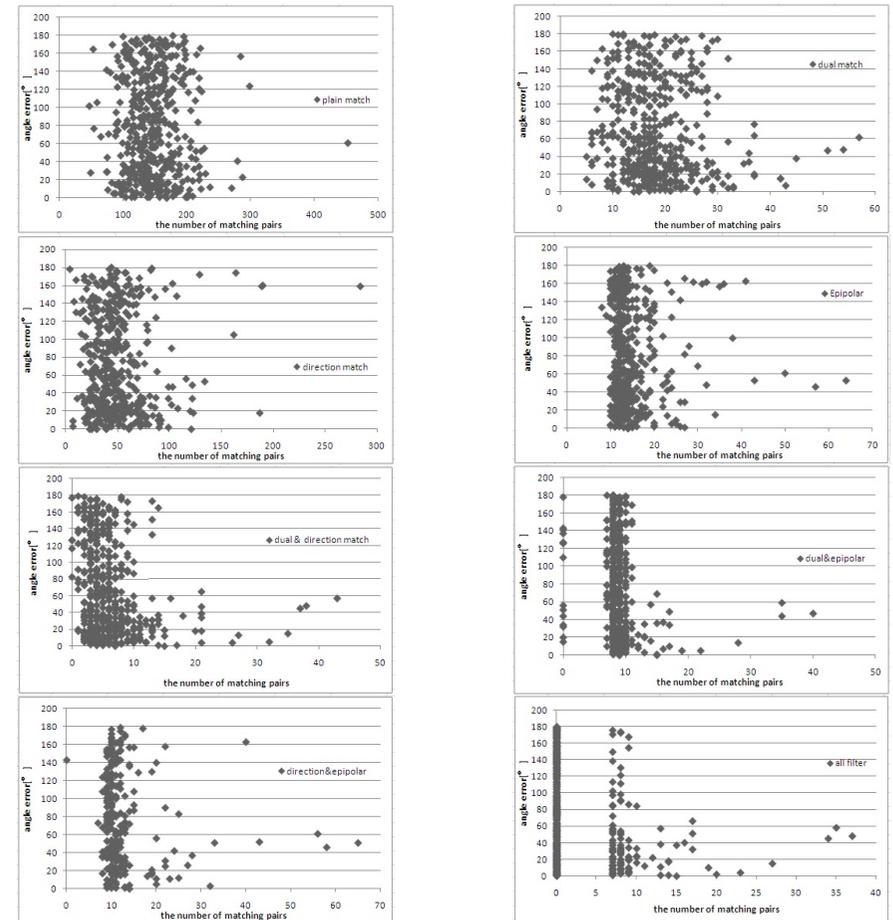


図 11 マッチングペアと誤差角度の分布 左上から単方向マッチ、その右が双方向マッチ、同様にその左下へ SIFT 方向、RANSAC、双方向と SIFT 方向、双方向と RANSAC、SIFT 方向と RANSAC、すべてのフィルタを使用した場合である。

Fig. 11 angle error and the number of matching pairs

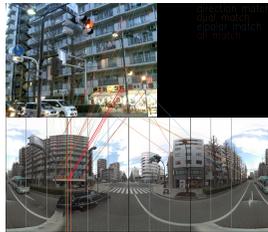


図 12 (東京都府中市若松町)  
(35.6702523169523  
139.494699221417)  
(103.87, 95.92(dir&epip))



図 13 (東京都調布市下石原)  
(35.655641201939  
139.536152663266)  
(268.18, 300.79(all))

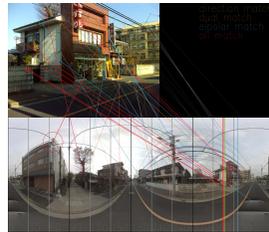


図 14 (都道 119 号線)  
(35.6555266583038  
139.536558167515)  
(268.18, 313.50(dual&dir))

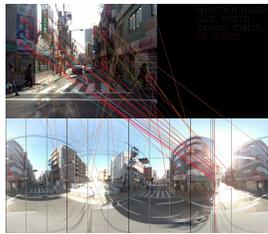


図 15 (都道 119 号線)  
(35.6535187557404  
139.542922531762)  
(322.46, 304.70(dual&dir))



図 16 (都道 119 号線)  
(35.6532533421309  
139.544068585323)  
(228.98, 213.87(dir))



図 17 (東京都調布市小島町)  
(35.6544052546377  
139.543638790123)  
(328.99, 329.84(epip))

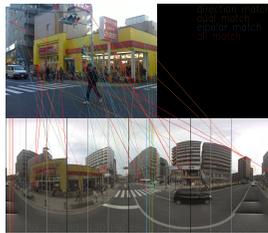


図 18 ((東京都日野市高幡)  
(35.6608168861767  
139.413442276399)  
(93.62, 37.06(dual&dir))

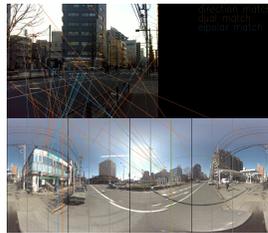


図 19 (東京都府中市若松町)  
(35.6697869108872  
139.495250202083)  
(309.33, 259.73(dual&dir))

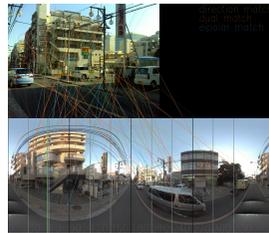


図 20 (東京都調布市小島町)  
(35.6538459431489  
139.541407316607)  
(183.17, 245.12(dual&dir))

図 21 マッチング点の対応例、図中の緑の線が SIFT 方向フィルタ、オレンジが双方向フィルタ、水色が RANSAC フィルタを示している。マッチペアは丸で対応しており、推定方向値は垂直線で描画されている。各図の下は (住所)(緯度、経度)(実方向、最良推定値 (手法名)) で構成されている

Fig. 21 examples of matching pairs



図 22 (ニューヨーク  
NY 10001, USA)  
(40.7514647222  
-73.9926783333333)  
(209.24,214.005(epip))

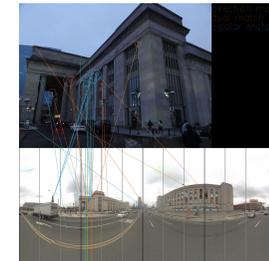


図 23 (フィラデルフィア  
PA 19104, USA)  
(39.95477138889  
-75.1829)  
(153.92, 105.909(dir))



図 24 (ニューヨーク  
NY 10001, USA)  
(40.750575  
-73.9925263889)  
(236.8, 201.701(dual))

図 25 国外でのマッチング点の対応例、ニューヨーク、フィラデルフィアでも利用できる  
Fig. 25 examples of matching pairs

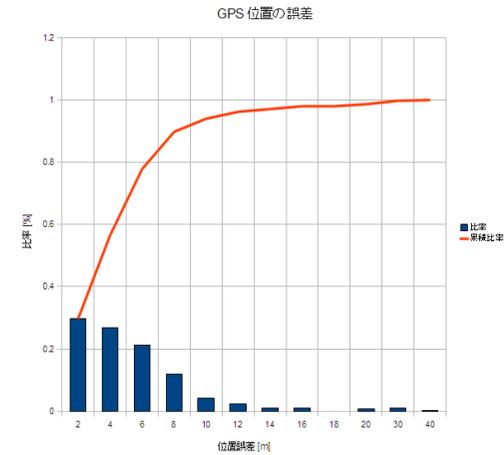


図 26 GPS の位置のずれの分布  
Fig. 26 Differences of the GPS position