# ジオタグ付きツイートを用いた交通路の抽出

谷　直樹† 風間　一洋† 榊　剛史†† 吉田　光男†††

† 和歌山大学 システム工学部　〒 640–8510 和歌山県和歌山市栄谷 930
†† 東京大学 工学系研究科　〒 113–6654 東京都文京区本郷 7–3–1
††† 豊橋技術科学大学 情報・知能工学系　〒 441–8580 愛知県豊橋市天伯町雲雀ヶ丘 1–1
E-mail: †{s161032,kazama}@sys.wakayama-u.ac.jp, ††sakaki@ipr-ctr.t.u-tokyo.ac.jp, †††yoshida@cs.tut.ac.jp

**あらまし**　Twitter の発言位置が付加されるジオタグ付きツイートを用いて，人間の移動や観光地情報の分析などの研究がさかんに行われている．しかし，東日本大震災時に提供された「通れたマップ」のような交通経路の抽出を考えると，散発的に行われるツイート時点しか位置が取得できないために単一ユーザの移動情報だけでは経路を再現できない上に，移動手段側ではなくユーザ側の位置であるためにツイート時の移動手段の利用を正確に判定できないという問題が存在する．そこで，投稿中又は前後に高速な交通手段を利用したと思われるツイートを抽出し，対象区域を細分化した各矩形領域内で近接している二つのツイートを Hough 変換することで交通路の断片と思われる近似直線を求め，それらをグループ化することで多くのユーザが利用した公共交通機関の交通路を抽出する方法を提案する．実際に，ジオタグ付ツイートアーカイブから関東の JR 山手線周辺区域と関西の JR 大阪環状線周辺区域から抽出した交通路を可視化する．さらに，各路線の位置と比較することで，手法の有効性を示すと共に，本手法の限界について分析する．

**キーワード**　Twitter, ジオタグ,Hough 変換, 交通路

## 1. はじめに

スマートフォンの普及に伴い，身の回りで起きた出来事などを簡単に発信できるマイクロブログサービスである Twitter(注1) が注目されている．特にユーザの発言位置が付加されるジオタグ付きツイートにおいては，ユーザの位置や移動を知ることができるだけでなく，ツイートの内容と組み合わせて分析することにより，例えば地震の震源地や台風の移動経路などの現実世界の状況を把握するためのソーシャルセンサーとして活用できることが知られている [1]．本稿では，ジオタグから得られるユーザの位置から，ある同一条件を満たす多くのユーザの位置を集積することで，それらの人たちが共通で利用した経路の検出を試みる．例えば，鉄道やモノレール，バスなどの公共交通機関は日常的に多くの利用者を運ぶことから，その経路上の Twitter ユーザの位置を集積・分析すれば，主な公共交通機関の交通路を推定できると考えられる．もちろん，このような静的な経路データは他の手段でも入手できるが，東日本大震災時に提供された「通れたマップ」のような震災直後に車が通行できた経路や，直後の停電や安全確保のための交通機関の運行停止による帰宅難民の行動の把握，または花見や紅葉の時期の名所の把握など，様々な状況下で動的に生成される経路の発見にも利用できる．さらに，推定された経路情報を用いて公共交通機関で移動中のユーザのツイートをタグ付けできれば，ユーザの訪問・滞在区域の分離や，目的の推定など，他の目的にも応用できると思われる．

ただし，ツイートは散発的にしか行われないために単一ユーザの移動情報だけでは経路を忠実に再現できない上に，移動手段側ではなくユーザ側の位置であるためにツイート時の移動手段の利用を正確に判定できないという問題が存在することから，既存の経路検出手法をそのまま適用することはできない．そこで，投稿中又は前後に高速な交通手段を利用したと思われるツイートを抽出し，対象区域を細分化した各矩形領域内で近接している二つのツイートを Hough 変換することで交通路の断片と思われる近似直線を求め，それらをグループ化することで多くのユーザが利用した公共交通機関の交通路を抽出する方法を提案する．

本稿の構成は次の通りである．2 章では，本研究と関連研究との差分について述べる．3 章では，交通手段使用時につぶやきを行ったと想定されるツイート群を抽出し，そのツイート群から近似直線として交通路を抽出する手法について述べる．また 4 章では，本手法の有効性を確認するために，実データを用いて路線上の近似直線がどの程度抽出できているかの評価を行う．最後に 5 章では，まとめと今後の課題について述べる．

## 2. 関連研究

### 2.1 ユーザの移動経路の抽出

ユーザの移動経路抽出に関しては様々な研究が存在する．

例えば，Hada らは，多数のプローブカー（GPS と通信機能を搭載した自動車）の位置を収集して地図上に可視化することで，震災時に通行可能な経路を抽出した [2]．また，山田らは，あらかじめ被験者から収集した移動経路情報を学習データとして保持し，空間的に同一となる移動経路の通過回数と，空間・時間的に同一である移動経路の通過回数に基づいて移動経路の抽出を行った [3]．また，石野らは，被災時のユーザ行動経路情

(注1)：https://twitter.com/

報を含むツイートをあらかじめ抽出し，移動元，移動先，移動手段を表すと考えられる単語に CRF 法も用いてタグ付けを行うことでユーザの移動経路を抽出する手法を提案した [4]．また，対象が移動経路ではないが，複数ユーザの履歴を集積する点が類似している手法として，大森らは，Flickr の写真とともに付与されることが多いタグと撮影位置には相関関係があり，タグは撮影位置の特徴を表していると仮定し，「beach」とタグ付けられた写真を抽出することで海岸線を描画する手法を提案した [5]．

これらの手法では，対象とする位置が常に抽出対象とする移動経路上にあったり，テキストやタグなどの情報を用いて抽出対象に直接関係する位置だけに絞り込めているのに対して，本手法ではまったく関係ない位置も多く含まれている状況下で，多くの人の位置を統合することで創出される経路の抽出を試みる点が異なる．

### 2.2 ユーザの移動手段の判定

類似研究として，ユーザの移動軌跡から，同一移動手段を用いた区間とその区間の移動手段（例，徒歩，車，電車など）を判定する移動手段推定の研究が行われている．

Reddy らは，ユーザが携帯する GPS 搭載機器が一定間隔で受信する GPS 軌跡データから計算される速度・加速度などの特徴から，移動手段を判定する分類器を構築した [6]．Leon らは，さらにユーザの位置を別途入手したバス停や駅などの位置を照合することで，速度だけでは判別が困難な電車・バス・自動車などに対しても移動手段を推定できることを示した [7]．Zheng らは，速度・加速度・移動距離に加えて，大きく変化している速度・進行方向の GPS 測位点や停止している GPS 測位点の割合を考慮し，移動形態の推定を行った [8]．また遠藤らは，GPS の移動軌跡から一旦軌跡画像を生成し，それを Deep Learning を用いて特徴を学習することで，移動手段を推定した [9]．

これらの手法は，複数の移動手段を用いることを前提としている点では本手法と同じである．しかし，これらの手法は比較的短い一定の時間間隔で取得したユーザの GPS 位置を利用できるのに対して，本手法はユーザの位置を取得できるのはツイート時だけであり，測定間隔が長すぎたり，同一人物の履歴だけでは測定点数が少なすぎるなどの問題が生じ，そのまま適用することは困難であることから，多くの人の位置を統合して判定する点が異なる．

## 3. 提 案 手 法

### 3.1 手法の概要

本稿で対象とするようなジオタグ付きツイートからの交通路抽出においては，いくつかの問題点が存在する．

（1） 位置取得タイミングの制御の問題．ツイート時にしか位置は取得できないので，取得タイミングはユーザの行動に依存する．通常は取得間隔が長い上に，タイミングも不定であり，個々のユーザのジオタグ付きツイートだけから移動経路を再現することはできない．

（2） 移動手段の位置取得の問題．取得できるのはユーザの位置だけなので，その位置が移動手段の移動経路上にあるとは限らない．

（3） ユーザの移動手段利用判定の問題．連続する二つのツイート間の移動速度から移動の有無は推定できても，ツイートした瞬間に移動していたかどうかは推定できない．

例えば，「通れたマップ」に関する Hada らのプローブカーを用いた研究では，これらの 3 条件をすべて満たしているのに対して，我々のジオタグ付きツイートを用いたアプローチでは，すべての条件が成り立たないという大きな前提条件の違いがある．そこで，これらの問題点に対応した方法が必要となる．

そこで，以下の交通路抽出法を提案する．

（1） ボットアカウントの除外
（2） 移動ツイート対の抽出
（3） 交通路の近似直線の抽出
（4） 同一経路と推定される近似直線のグループ化
（5） 抽出結果の可視化

### 3.2 ボットアカウントの除外

Twitter ボットは，自動的にツイートするプログラムである．設定した文章を好きな時間に自動でツイートするボットを簡単に作成できるボット生成サービスも多く，広く利用されている．この中にはジオタグの位置を設定できるボットも存在するが，人間のつぶやきだけを取り出すために，分析前にボットアカウントを除去する必要がある．

一般的に，利用クライアント名（source 値），ユーザ名（screen_name 値），プロフィール情報（description 値）でボットアカウントであることを明示することが多いことから，これらの情報に「BOT」,「Bot」,「bot」などの単語が含まれている場合に処理対象から除外する．

### 3.3 移動ツイート対の抽出

自動車，バス，電車，新幹線などの別の人が運転する交通手段ではツイート可能であるが，ツイート時にそれらの交通手段で移動していたかどうかは判別できない．

そこで，ユーザの連続する二つのツイートの投稿位置と時間から求めた移動速度が閾値 $T_v$ 以上の場合に上記のような交通機関により移動したとみなす．本稿では，この二つのツイートを**移動ツイート対**，各ツイートを**移動ツイート**と呼ぶ．

ただし，すでに述べたように，移動ツイートの集合がそのまま交通路を表すわけではない．例えば，近鉄大阪線と JR 大阪環状線が交差する鶴橋駅周辺の移動ツイートを，Google Maps 上にマーカで表して可視化した結果を，図 1 に示す．なお，図中の黒い部分は，黒いエッジ部を持つマーカが密集することによって生じる．確かに近鉄大阪線と JR 大阪環状線に対応する縦と横の黒い線状の軌跡が 2 本観測でき，中央部でそれらが交差している地点が鶴橋駅である．しかし，近鉄大阪線の左部分では，移動ツイート量の不足により黒い部分が観測されず，逆に図の中央下部の各路線とは関係のない部分に黒い部分が観測されている．また，大阪市の主要な繁華街・オフィス街であり複数路線の乗り換え駅である難波駅や梅田駅周辺では，このような路線の目視による判別が不可能なほど大量の移動ツイートが広範囲に渡って密集している．つまり，移動ツイートをそのまま扱うだけでは，交通路の判別が困難なことがわかる．

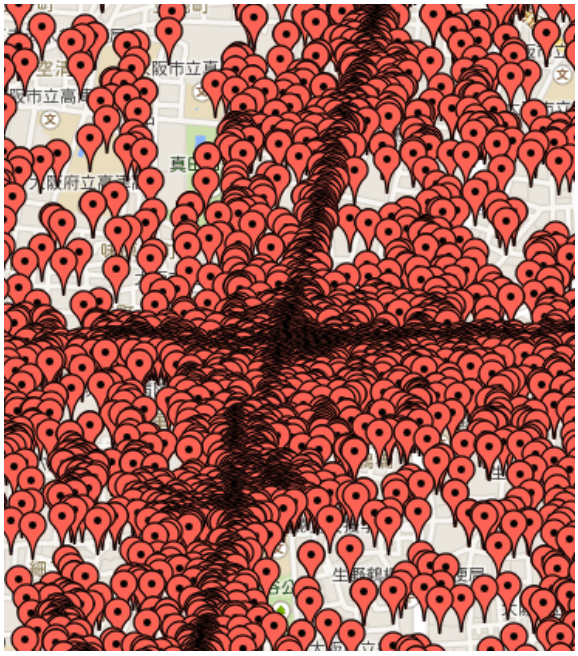

図 1　鶴橋駅周辺の移動ツイートの可視化例

なお，ここで移動ツイートを対として抽出する目的は，ツイート前後に交通機関による移動を行ったかどうかを判定するためだけである．その後の処理では再び分解して，独立した移動ツイートとして扱う．

このステップの処理を一般化すれば，ある条件に合致するツイート群だけを抽出することである．そこで，移動速度以外の条件判定を行えば，他の目的にも適用できる．例えば，本文に「花見」,「桜」などの単語を含むツイートだけを取り出せば，花見に適した桜並木などを抽出できると思われる．

### 3.4　交通路の近似直線の抽出

次に，対象地理空間を複数の矩形領域に分割し，各矩形領域内の移動ツイート群の特に密度が高い連続部分を **Hough 変換** [10] を用いて直線として抽出することで，交通路を部分的に近似する直線を抽出する．

Hough 変換は，画像から得られた多くのエッジを，原点から垂直に引いた直線の距離と角度の空間（Hough 空間）上に写像し，パラメータ頻度が高い箇所を再び元の空間上に逆写像することで，エッジ群を通る直線を抽出する手法である．原点からの距離を $\rho$，角度を $\theta$ とすると，以下の式 1 が成り立つ．

$$\rho = x\cos\theta + y\sin\theta \tag{1}$$

ただし，移動ツイートでは，すでに述べたように前後に移動したかどうかを推定できるだけで，必ずしも交通路上にあるとは限らない．例えば，自宅でツイートしてから電車で移動し，待ち合わせをしていたレストランで再びツイートした場合には，どちらのツイートの位置も交通路とは離れた場所となる．このような場合は移動ツイート対を繋ぐエッジは交通経路とはかけ離れた位置・角度を持つために，Hough 変換の元データとしてそのまま使うことはできない．

そこで，移動ツイート対を一旦分解してから，近接する二つの移動ツイートを繋いでエッジとすることで得られるエッジ集合を入力データとする．ここで，エッジを生成する二つの移動ツイートは，必ずしも同一ユーザのものではなく，エッジも交通路上に乗っているとは限らない．つまり，入力データの段階では通常の Hough 変換と違って大量のノイズが混在しているが，パラメータ頻度が高い箇所だけを逆変換することで，ノイズが除去されることになる．

各矩形領域内の近似直線を抽出するアルゴリズムは以下の通りである．

（1） 閾値 $T_d$ 以下の距離の二つのツイートを通過するエッジ群を求める．

（2） エッジ群を Hough 変換し，Hough 空間上の距離・角度に対して分割された領域ごとのエッジ頻度を集計する．

（3） Hough 空間上のエッジ数が最大となる領域の距離，角度の平均値を求めて，地理空間上に直線として逆写像する．

（4） 矩形領域内のユーザ数と集中度が，閾値 $T_u$ と $T_c$ を下回る場合は，ノイズとみなして処理対象から除外する．

なお，Hough 空間のユーザ数が少ない領域は，特定のユーザが繰り返し訪問する自宅や職場であることが多い．そこで矩形領域内のユーザ数が閾値 $T_u$ を下回る場合には，処理対象から除外することとする．

また，集中度は矩形領域内の分布が特定の部分に偏っているかどうかを表す指標であり，移動ツイートが密集しているほど大きくなり，均一に分散している場合は小さくなる．矩形領域 $(i,j)$ 内の集中度 $c_{i,j}$ は，移動ツイート数 $t_{i,j}$ とエッジ数 $e_{i,j}$ を用いて，以下のように定義する．

$$c_{i,j} = \frac{e_{i,j}}{t_{i,j}} \tag{2}$$

今回は，集中度が閾値 $T_c$ を超える場合のみを対象とした．

### 3.5　同一経路と推定される近似直線のグループ化

上記の処理では，独立した短い直線群として抽出される．しかし，電車の路線などの経路は，連続した長い直線群であることから，同一経路上にあると思われる近似直線をグループ化する．

矩形領域 $(x,y)$ の近似直線 $l$ と，同一経路上にあると推測される近似直線 $l'$ を発見する概念図を図 2 に示す．ここで，$l$ と $l'$ の中線を結ぶ線分を引き，$l$ と水平軸のなす角度を $\alpha$，$l'$ の角度を $\beta$，中線の角度を $\gamma$ とする．これらの角度が，電車や道路などの曲率を考慮した範囲差であれば，近似直線 $l$ と $l'$ は同一経路集合に属するとする．

ただし，GPS による位置計測や通信に支障があるなどの理由で位置が取得できない又は不正確だったり，乗客数が少ないなどの理由で充分なデータが得られない場所もあると考えられる．そこで，グループ化時にある程度のギャップは許容することとする，実際には，矩形領域 $(x,y)$ の近似直線 $l$ の相手の近似直線 $l'$ を探索する範囲を，$(x-1,y-1),(x+1,y-1),(x-1,y+1),(x+1,y+1)$ の範囲の 8 個の矩形領域から，$(x-2,y-2),(x+2,y-2),(x-2,y+2),(x+2,y+2)$ の範囲の 24 個の矩形領域に拡大する．このように探索範囲を拡大することで，例えば隣接している 8 矩形領域に近似直線が見つからない場合でも，さらにその先の 16 矩形領域に近似直線が

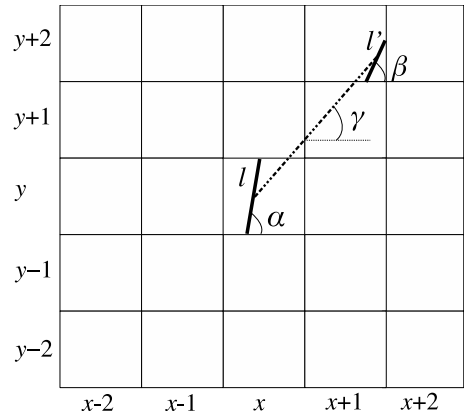


図 2　近似直線同一経路集合を構築する例

存在すれば，それらをグループ化することができる．

さらに，具体的なアルゴリズムを以下にしめす．

（1） 指定された範囲から，未処理の近似直線 $l'$ を一つ取り出す．未処理の近似直線がない場合は終了する．

（2） $diff(\alpha,\beta) \leqq T_{\theta_1}$ を満たさなければ，(1) に戻る．

（3） $diff(\alpha,\gamma) \leqq T_{\theta_2} \wedge diff(\beta,\gamma) \leqq T_{\theta_2}$ を満たさなければ，(1) に戻る．

（4） 直線 $l'$ を同一経路グループに追加して，(1) に戻る．

なお，$l$ の角度を $\alpha$，$l'$ の角度を $\beta$，$l$ と $l'$ の中線の角度を $\gamma$，$diff(\theta,\theta')$ は，角度 $\theta$，$\theta'$ の角度差 $\theta''(0 \leqq \theta'' < 360)$ を求める関数，$T_{\theta_1}$，$T_{\theta_2}$ は角度の閾値とする．抽出された近似直線にさらに同じアルゴリズムを提供することで，同一経路と思われる近似直線グループが抽出できる．

なお，グループ化後の近似直線グループのサイズが閾値 $T_g$ より小さい場合には，明確な交通経路ではないノイズと考え，除外する．

### 3.6　抽出結果の可視化

本手法に基づいて抽出された近似直線のグループを可視化するシステムを作成した．ユーザが Web ブラウザから可視化システムにアクセスすると，Google Maps API を用いて地図上に近似直線を可視化する．このプログラムは Python で記述され，Apache Web Server の CGI として動作する．

処理時間を短縮するために，一旦移動ツイートのみを抽出し，そのユーザ識別番号やツイート位置（緯度・経度）などの Twitter API で取得できる情報と，事前に計算したツイートが投稿された矩形領域番号などを MongoDB に格納した．ただし，本手法の各過程のパラメータを変更したり，各段階における処理の有効性を確認できるように，交通路の近似直線の抽出と同一経路と推定される近似直線のグループ化は実行時に処理している．

## 4. 評　　価

### 4.1　ツイートデータセット

Twitter Streaming API(注2) を用いて，ジオタグが付いたツイートだけを収集し，JSON 形式で保存したツイートデータセットを作成した．この中から，2013 年 11 月 1 日〜4 月 31 日の 6ヶ月分を抽出して，評価に使用した．データに含まれるツイート数は 67,619,243，ユーザ数は 1,130,328 である．

### 4.2　経路の抽出

関東の JR 山手線周辺区域と，関西の JR 大阪環状線周辺区域の二つの区域に対して，交通路抽出を行った．各領域の面積は，JR 山手線周辺区域が 200km$^2$，JR 大阪環状線周辺区域が 100km$^2$ であった．

移動ツイート対の抽出における速度の閾値 $T_v$ は，想定した移動手段で最低速度と考えられる各駅停車の速度である 18km/h とした．実際には各駅停車の移動速度は 24km/h の区間が多かったが，18km/h 区間も対象にできるように，低い値を閾値とした．この結果，JR 山手線周辺区域から 438,175，JR 大阪環状線周辺区域から 156,450 の移動ツイートが抽出された．

交通路の近似直線の抽出に関しては，対象領域は 250m の矩形領域に分割し，Hough 空間は，距離の軸では 100m，角度の軸では 10 度ごとに分割したさらに，ツイート間距離の閾値 $T_d$ を 100m，ユーザ数の閾値を $T_u$ を 3，集中度の閾値 $T_c$ を 0.7 とした．

同一経路と推定される近似直線のグループ化においては，角度差の閾値 $T_{\theta_1}$ と $T_{\theta_2}$ を共に 30 度とした．さらに，同一経路近似直線グループのサイズの閾値 $T_g$ を 3 とした．

### 4.3　抽出結果の可視化

JR 山手線周辺区域と JR 大阪環状線周辺区域の可視化結果を，それぞれ図 3 と図 4 に示す．どちらも JR 山手線と JR 大阪環状線は比較的忠実に抽出されているが，図 3 の東京駅や新宿駅，図 4 の梅田駅や難波駅のような主要駅周辺と環状線の内側は，ノイズと思われる短い近似直線が抽出されてしまっていることがわかる．

ここで，移動ツイートをそのまま可視化した図 1 と比較すると，図の中央にある鶴橋駅の左部分の移動ツイート量が少ない矩形領域に関しては，図 4 の該当部分（右半面中央部）を見れば近鉄大阪線に沿って近似直線が抽出できている．しかし，鶴橋駅の下側では，近似直線が抽出されなかったり，路線との誤差が多い矩形領域が存在している．

なお，現時点では矩形領域内において近似直線を 1 本しか抽出できない制約があるので，複数の路線が並行又は交差する矩形領域では路線が途切れる現象として現れている．また，駅近傍に多くの人が集まる施設がある場合には，ノイズと思われる短い近似直線や，路線から外れた近似直線が抽出されている．

### 4.4　ノイズ除去の評価

まず，本手法の各段階でノイズがどのように除去されるかを，可視化結果に基づいて評価する．

本手法で抽出した JR 大阪環状線の野田駅周辺の近似直線を，ユーザ数と集中度によるノイズ除去を行わずに可視化した結果の一部を，図 5(a) に示す．移動ツイートをそのまま可視化した図 1 と比較すると，地図上の電車の路線のほとんどの部分に対して連続した近似直線が抽出されている反面，電車の路線とは関係ない近似直線も多く抽出されていることがわかる．

次に，ユーザ数と集中度によるノイズ除去を行った可視化結

(注2)：https://dev.twitter.com/docs/api/streaming

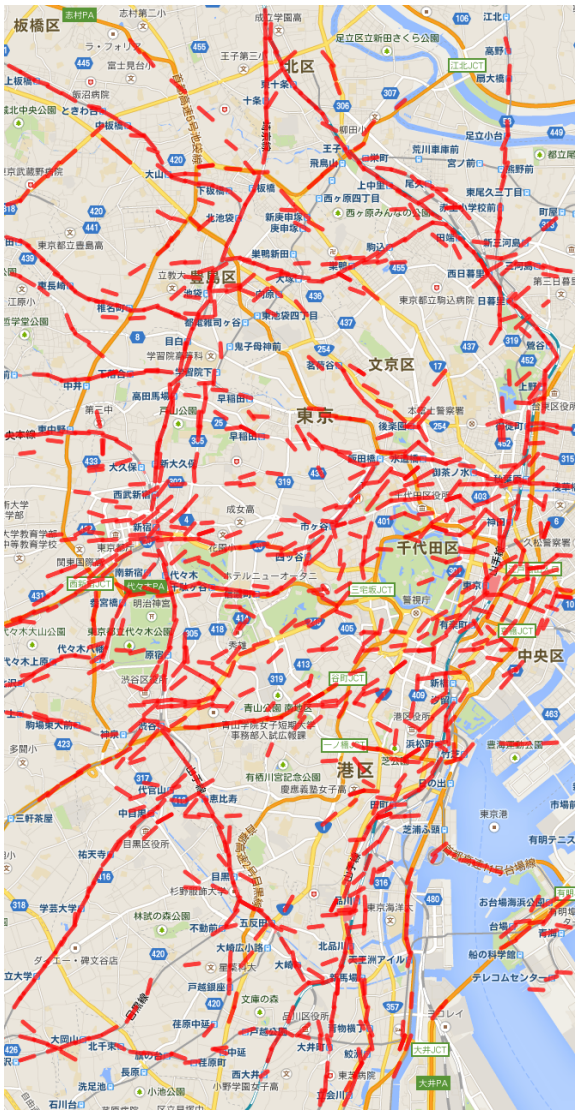


図 3　JR 山手線周辺区域の可視化結果

果を，図 5(b) に示す．この結果から，電車の路線とは関係のない近似直線がほとんど除去されているが，まだ一部ノイズと思われる近似直線が残っていることがわかる．

さらに，同一経路と推定される近似直線をグループ化してから小さなグループを除去した後の最終的な抽出結果を，図 5(c) に示す．孤立した近似直線，又は短い経路を構成する近似直線グループを除去することで，長く連続した経路を構成すると思われる近似直線グループに絞り込まれていることがわかる．

図 5 という範囲の狭い領域だけでなく，より広い領域でノイズ除去が有効に働いていることを証明するために，さらに JR 山手線周辺区域と JR 大阪環状線周辺区域において，上記の 3 段階における近似直線数の変化を調べた．その結果，JR 山手線周辺区域では 2110 本，1396 本，1048 本，JR 大阪環状線周辺区域では 998 本，588 本，415 本となった．この結果から，交通路上にない近似直線のかなりの割合を，ノイズとして除去できたと思われる．

### 4.5　評価用路線データ

抽出結果を評価する際に正解とする路線データとして，国土交通省が全国総合開発計画，国土利用計画，国土形成計画などの国土計画の策定や実施の支援のために作成した国土数値情報の中から，鉄道時系列データ (注3) を用いた．鉄道時系列データは，XML ベースのマークアップ言語である GML を用いた地理情報標準プロファイル（JPGIS）第 2.1 版を用いて記述され，鉄道，路線，駅に関する情報を含んでいる [11]．

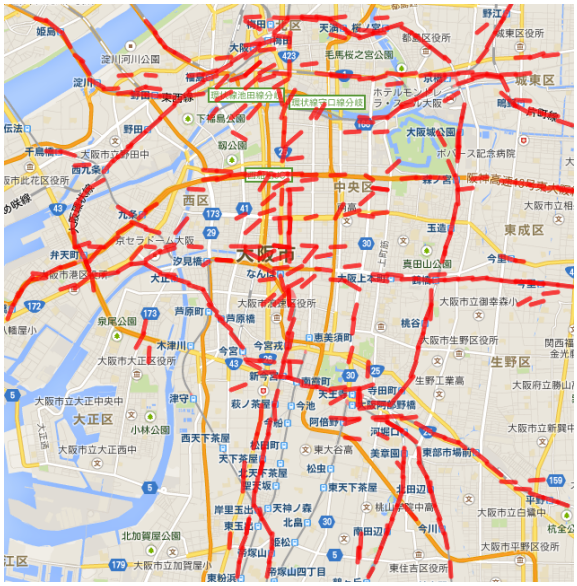


図 4　JR 大阪環状線周辺区域の可視化結果

表 1　鉄道時系列パッケージの代表的なタグ

| クラス | 属性・関連役割 | タグ名 | 型 |
|---|---|---|---|
| 鉄道 | 事業者種別 | int | 事業者種別コード |
| | 路線名 | lin | string |
| | 運営会社 | opc | string |
| 路線 | 路線 | loc | CurvePropertyType |
| 駅 | 地点 | loc | PointPropertyType |
| | 駅名 | stn | string |

代表的なタグを，表 1 に示す [12]．型 CurvePropertyType は複数地点の緯度・経度のリストであり，型 PointPropertyType は 1 点の緯度・経度である．なお，Twitter のジオタグでは WGS84 測地系を用いているのに対して，鉄道時系列データでは JGD2000 測地系を用いていることから，位置の誤差が生じることが考えられるが，実際にはその差は数 cm〜m 程度であることから，そのまま使用した．

さらに，opc タグと lin タグを手掛かりに路線を判定し，さらに loc タグから路線の緯度・経度のリストを取得した．なお，lin タグの路線名は一般的に使われる路線名と 1 対 1 対応しているわけではなく，日比谷線のように「2 号線日比谷線」となっていたり，山手線のように，区画ごとに「山手線」,「東海道線」,「東北線」と複数に分割されていた．そこで，実際の路線名と鉄道時系列データは，人手で対応付けを行った．

### 4.6　路線の再現性の評価

一般に，交通路の再現性は，対象領域の Twitter アクティブユーザ・移動ツイート数と，通信と位置取得に用いる GPS 衛星・携帯基地局・Wi-Fi アクセスポイントの電波強度と補足数

(注3)：http://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N05.html


(a) ノイズ除去を行わない可視化結果


(b) ユーザ数と集中度によるノイズ除去を行った可視化結果


(c) 本手法の可視化結果

図 5　ノイズ除去の評価

に大きく左右されると考えられる．また，本手法においては，地理空間上を静的に領域分割を行っているため，路線の距離が短い領域では，路線上の移動ツイート数が少なくなり，近似直線を抽出できないといった問題がある．そこで，本手法で各路線がどの程度再現できているかを評価する．

具体的には，まず JR 山手線周辺区域を本手法と同様に矩形領域に分割した上で，評価用路線データに含まれる各路線を表す地点のリストが存在している矩形領域の集合を求めて，その領域において本手法で抽出した近似直線がどの存在するかどうかを評価する．この評価には，次の再現率 $R$ を用いる．

$$R = \frac{P}{C} \tag{3}$$

ここで，$C$ は鉄道時系列データを元に判断した各路線が実際に通過している矩形領域数であり，$P$ はそれらの矩形領域の中で本手法によって最終的に近似直線が抽出された矩形領域数である．

ただし，現時点では，本手法は一つの矩形領域で一つの近似直線しか抽出しないという制約があるために，近似直線が抽出されていても，方向が異なる別路線のものである可能性があるが，ここでは評価方法を簡略化するために，その確認はおこなわない．また，鉄道時系列データは路線をそれが通過する地点のリストで表現するために，路線が通過している矩形領域が評価の対象にならない場合も存在するが，そのような領域はごくわずかであったために，今回は特に配慮しない．

評価対象とした路線名に関するデータと再現率 $R$ を表 2 に示す．この結果を見ると，地上路線は全体的に再現率が高い．この理由としては，今回評価対象とした JR 山手線周辺区域は人口 1000 万人以上であることからジオタグ付きでツイートするユーザ数が充分得られること，昼間の交通渋滞と電車網の普及から電車の利用率が高いこと，東京都心部の地価や家賃の高さから郊外部と都心部を往復するユーザが多いことなどが考えられる．ただし，地下鉄路線においては全体的に再現率が低い．この違いが出てくる理由は，地上では GPS 衛星の捕捉が可能で，さらに複数の Wi-FI 基地局位置取得が可能であるのに対して，地下鉄路線では位置取得に何らかの制約や問題があるからだと考えられる．

さらに，各路線ごとの詳細な分析を次に示す．

#### 4.6.1　山　手　線

山手線の再現率は 0.931 と一番高かった．例えば，JR 東日本が公開している 2013 年度の各駅の 1 日平均の乗車人員のデータを調べると，山手線で一番多い新宿駅が JR 東日本管区内で 1 位で 751,018 人/日，一番少ない鶯谷駅で JR 東日本管区内で 162 位で 24,481 人/日であった (注4)．山手線のほとんどの駅は複数の路線の乗換駅であることも考慮すると，鶯谷駅であっても乗降客数が少ないだけで，乗車したまま通過する客数は膨大であると考えられ，充分なアクティブユーザ数・移動ツイート数が確保できたと考えられる．

#### 4.6.2　京　王　線

京王線の再現率は 0.823 と，地上路線の中では比較的低かった．再現性が低かった矩形領域は新宿駅から笹塚駅の間であり，この間は京王線と京王新線の二つの路線が平行して走っていた．このように，一つの領域の中に複数の路線が存在する場合には，

(注4)：http://www.jreast.co.jp/passenger/

表 2　評価する路線データと再現率 $R$

| | 路線名 | 1 駅あたりの平均乗降客数 / 日 | opc タグ | lin タグ | 正解領域数 $C$ | 本手法での領域数 $P$ | 再現率 $R$ |
|---|---|---|---|---|---|---|---|
| 地上路線 | 山手線 | 447,640 | 東日本旅客鉄道 (旧国鉄) | 山手線，東海道線，東北線 | 153 | 142 | 0.928 |
| | 中央線 | 47,527 | 東日本旅客鉄道 (旧国鉄) | 中央線 | 59 | 53 | 0.898 |
| | 東横線 | 106,149 | 東京急行電鉄 | 東横線 | 30 | 27 | 0.900 |
| | 小田原線 | 66,543 | 小田急電鉄 | 小田原線 | 21 | 19 | 0.904 |
| | 京王線 | 53,196 | 京王帝都電鉄<br>京王電鉄 | 京王線 | 17 | 14 | 0.823 |
| | 目黒線 | 49,854 | 東京急行電鉄 | 目黒線 | 22 | 14 | 0.636 |
| 地下鉄路線 | 日比谷線 | 63,055 | 東京地下鉄<br>帝都高速度交通営団 | 2 号線日比谷線 | 76 | 47 | 0.618 |
| | 千代田線 | 74,466 | 東京地下鉄<br>帝都高速度交通営団 | 9 号線千代田線 | 77 | 45 | 0.584 |
| | 銀座線 | 56,473 | 東京地下鉄<br>帝都高速度交通営団 | 3 号線銀座線 | 60 | 49 | 0.816 |
| | 有楽町線 | 40,000 | 東京地下鉄<br>帝都高速度交通営団 | 8 号線有楽町線 | 72 | 32 | 0.444 |

移動ツイートが分散して集中度 $c_{i,j}$ の閾値を超えなかったと思われる．

#### 4.6.3　目　黒　線

目黒線の再現率は 0.636 と，地上路線の中で一番低かった．再現性が低かった矩形領域は不動前駅から洗足駅の間であり，この領域では 2006 年 7 月に地下化工事が完了しており，後述する地下鉄と同じ理由で再現性が低かったのではないかと思われる．なお，地下化がおこなわれていない区間では，再現性が高かった．

#### 4.6.4　銀　座　線

銀座線の再現率は 0.816 と，地下鉄路線の中では一番良い結果を示していたが，詳しく調べると，近似直線の集合が銀座線の路線を再現していなかった．これは，今回評価に用いた再現率は単にその路線が通過している各矩形領域内で近似直線が抽出されたかどうかで判定していて，路線を忠実に再現しているかどうかまでは踏み込んでいないからである．そこで，例えば，近傍の山手線や中央線の近似直線を誤判定したり，青山通りや中央通りなどの交通量が多い国道の渋滞状況を報告するツイートなどにより一部が再現されたのでないかと思われる．

#### 4.6.5　有 楽 町 線

有楽町線は，全般的に再現率が低い地下鉄路線の中でも，0.444 と特に悪かった．詳しく調べると，再現された矩形領域であっても，実際には他路線の近似直線であった．ただし，有楽町線は銀座線のように近傍に他の路線や，高速道路，主要国道がほとんど存在していなかっただけで，銀座線と有楽町線はどちらも再現性が悪いという点では共通していると思われる．

### 4.7　地下鉄路線の位置取得精度の分析

地下鉄路線で全般的に再現率が低い理由として，スマートフォンによる位置取得に何らかの制約か問題があるからだと考えられる．

この原因を探るために，大阪の地下鉄である堺筋線と御堂筋線に実際に乗車して，ソフトバンクの iPhone の Twitter アプリを用いて駅及び駅間でツイートし，ツイートのジオタグの有無と取得位置の正確さを調べた．ツイートに付加された位置を Google Maps API を使って可視化した結果を，図 6 に示す．


(a) 都心部の位置取得状況　　(b) 郊外の位置取得状況

図 6　地下鉄の位置取得

ツイートのジオタグの有無に関しては，地下鉄乗車中に発言した 107 ツイート中，ジオタグは 88 ツイート（82.2%）に添付されていた．ただし，位置が添付されていない地点には駅間だけでなく駅付近も含まれ，ツイートに位置が添付されない原因は不明である．

取得位置の正確さに関しては，駅間で何回もツイートしているにもかかわらず，都心部では図 6(a) のようにツイート地点の代わりに近傍の駅の位置が取得され，郊外部では図 6(b) のように路線から大きく離れた位置が取得されていた．つまり，地下鉄路線の再現率が低かった理由は，経路上に広く分散しないために Hough 変換の対象エッジとして抽出されなかったことと，路線から離れた位置が取得されるためにノイズとして処理されたからだと考えられる．

今回実験に用いた iOS では Core Location フレームワークを用いて位置を取得するが，この内部では Bluetooth (iBeacon),

Wi-Fi，GPS，携帯などの複数の位置取得手段を使い分けている [13]．一般に位置取得精度は，複数の基地局・衛星位置から推定する Bluetooth や Wi-Fi，GPS が高く，基地局のセル（通信可能領域）だけで判定する携帯が低い．

つまり，取得された位置が局所的に集中する理由は，地下鉄では，iBeacon が設置されていない，衛星を補足できないだけでなく，位置推定に必要な複数の Wi-Fi 基地局が存在しないために，携帯のセルだけから判定したことで携帯基地局の位置が取得されたと考えられる．

また，都心部では路線上の位置が，郊外部では路線から離れた位置が取得される理由は，都心部では駅に基地局が設置されているのに対し，郊外部では駅には地上の携帯基地局の電波を地下に中継する機能しか持たない中継機が設置されているからではないかと思われる．

ただし，東京メトロ地下鉄や都営地下鉄では 2014 年 12 月から共通無線 LAN サービスが開始されるなど，今後 Wi-Fi 基地局の設置数が増加することで，今後は地下鉄においても正確に位置を取得できるようになることが期待できる．

## 5. おわりに

本稿では，ジオタグ付きツイートの中に含まれる，特に何らかの交通手段を利用している時につぶやいたと考えられる移動ツイートから，分割された矩形領域ごとに交通路の近似直線を抽出する手法を提案し，実際に JR 山手線周辺区域と JR 大阪環状線周辺区域の処理結果を可視化することで，公共交通機関の交通路が抽出できることを示した．

評価では，まずノイズ除去の効果を可視化・分析して，本手法の各段階の効果を確認した．次に，JR 山手線周辺区域の各路線ごとにどの程度再現できているかを評価し，各路線で再現性が高い又は悪い環境条件を分析した．その結果，地上路線は全般的に再現性が高いが，地下鉄路線では全般的に再現性が低いことが判明した．最後に，実際に地下鉄路線でジオタグ付きでツイートする実験を行い，ジオタグの有無と得られた位置の精度を調べて，地下鉄路線における位置取得状況と確認するとともに，再現性が低くなる技術的な根拠を考察した．

今後の課題の一つは，交通路を単なる直線集合ではなく，一本の曲線として抽出することである．そのために，矩形領域内で複数の近似直線を抽出することで断続箇所を減少させると共に，地下化や乗降客数が少ないなどの原因から生じる断続箇所を補完し，近似直線群をベジエ曲線などで近似することが必要である．

もう一つの課題は，今回の交通路の再現率に加えて，位置の精度を評価することである．

## 謝　辞

本研究は JSPS 科研費 26330345 の助成を受けた．

## 文　献

[1] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users:real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*, pp. 851–860, 2010.

[2] Yasunori Hada, Takeyasu Suzuki, and Itsuki Noda. Utilization of probe vehicle information in disasters in japan. In *Proceeding of the 15th World Conference on Earthquake Engineering (WCEE2012)*, 2012.

[3] 山田直治, 礒田佳徳, 南正輝, 森川博之. GPS 搭載携帯電話を用いた移動経路履歴に基づく訪問地・経由地予測システム. 情報処理学会研究報告 ユビキタスコンピューティングシステム (UBI), Vol. 2010-UBI-27, No. 4, pp. 1–8, 2010.

[4] 石野亜耶, 小田原周平, 難波英嗣, 竹澤寿幸. Twitter からの被災時の行動経路の自動抽出および可視化. 第 18 回年次大会, pp. 907–910. 言語処理学会, 2012.

[5] 大森雅己, 廣田雅春, 石川博, 横山昌平. Flickr は海岸線を描けるか？ 第 6 回データ工学と情報マネジメントに関するフォーラム 2014 (DEIM2014), 2014.

[6] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, No. 2, 2010.

[7] Leon Stenneth, Ouri Wolfson, Philip S. Yu, and Bo Xu. Transportation mode detection using mobile phones and gis information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 54–63, 2011.

[8] YU Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying MA. Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web (TWEB)*, Vol. 4, No. 1, pp. 1:1–1:36, 2010.

[9] 遠藤結城, 数原良彦, 戸田浩之, 小池義昌. 移動手段判定のための表現学習を用いた GPS 軌跡からの特徴抽出. 第 7 回 Web とデータベースに関するフォーラム (WebDB Forum 2014), 2014.

[10] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, Vol. 15, No. 1, pp. 11–15, 1972.

[11] 国土交通省国土地理院. 地理情報標準プロファイル（JPGIS）Ver. 2.1, 2009.

[12] 国土交通省国土政策局. 国土数値情報（鉄道時系列）製品仕様書 第 1.2 版, 2014.

[13] Apple. 位置情報とマッププログラミングガイド, 2014.