移動速度条件を考慮したジオタグ付きツイートからの交通路の抽出と分析

谷 直樹[†, a]　風間 一洋[†, b]　榊 剛史[††, c]　吉田 光男[†††, d]　斉藤 和巳[††††, e]

† 和歌山大学システム工学部 †† 株式会社ホットリンク ††† 豊橋技術科学大学情報・知能工学系
†††† 静岡県立大学経営情報学部

a) *s161032@center.wakayama-u.ac.jp* b) *kazama@ingrid.org* c) *sakaki@ipr-ctr.t.u-tokyo.ac.jp*
d) *yoshida@cs.tut.ac.jp* e) *k-saito@u-shizuoka-ken.ac.jp*

概要　交通手段によって，移動速度や移動距離などの特性の違いが存在する．本稿では，入力とするツイート集合の移動速度と移動距離の条件を変更することで，異なる特性の交通手段の経路を抽出しわける手法を提案する．まず，指定された移動速度と移動距離の条件に基づいて，Twitter ユーザの連続ツイートの投稿位置と時間から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．次に，その近似直線群を Google Maps 上に可視化した結果を分析し，異なる交通手段の経路が抽出されていることを示す．

キーワード　Twitter, ジオタグ,Hough 変換, 交通路, 速度域

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を簡単に発信できる Twitter が注目されている．特に位置情報が付加されたジオタグ付きツイートは，ユーザの発言場所や移動経路がわかるだけでなく，発言内容と組み合わせて分析することで，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 [1].

我々は，ジオタグ付きツイート中で，指定した条件を満たすツイート集合から，それらのユーザが共通で利用した移動経路を推定する手法を提案し，実際に日常的に多くの利用者を運ぶ公共交通機関の交通路の抽出に適用した [2]．なお，ユーザの散発的なツイートでは経路を忠実に再現するには不十分であり，ユーザの位置しか取得できないために交通機関の経路だけを切り出せないことから，既存の経路抽出法をそのまま適用できない．そこで，我々は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手法を提案した．

この手法は，目的に応じた条件を満たすツイート集合を与えば，その目的に合致した経路を発見できるように，汎用的に設計されている．例えば，移動中に呟いたツイート集合を与えれば移動経路を，4 月初旬の「花見」が含まれるツイート集合を与えれば桜並木を抽出することを想定している．

Copyright is held by the author(s).
The article has been published without reviewing.

本稿では，分析対象とする交通手段の移動特性に合わせて入力とするツイート集合を作成する条件を変更することで，目的に応じて経路を抽出しわける手法を提案する．例えば，交通手段によって移動速度が異なるだけでなく，渋滞が発生するなどの理由で移動速度が大きく変動する場合がある．また，飛行機や高速道路，新幹線などのような長距離移動に用いる交通手段もあれば，車やバスのように短距離移動に用いる交通手段もある．そこで，移動ツイートを抽出する際の移動速度と移動距離の条件を変えることで，実際に異なる交通手段の経路が抽出できるかどうかを調べる．

2　ジオタグ付きツイートからの交通路の抽出

2.1　ジオタグ付きツイートを用いる場合の問題点

例えば，東日本大震災直後に車が安全に通行できた経路を示したサービスである「通れたマップ」は，Hada らの GPS と通信機能を搭載したプローブカーを用いて収集した情報を分析する研究 [3] に基づいている．しかし，本稿のようにジオタグ付きツイートから交通路を抽出する場合には，これらの既存研究にない次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．位置の取得タイミングはユーザのツイート行動に依存する．通常は取得間隔が長い上に，タイミングも不定なので，個々のユーザのジオタグ付きツイートだけから交通路を再現できない．
2. 移動手段の位置取得の問題．交通機関ではなくユーザの位置しか取得できないので，常に交通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．

そこで，以下の手順で構成される交通路抽出法を用いる．

2.2 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り，設定した文章を自動でツイートするボット生成サービスも多く利用されている．ジオタグを付加してツイートするボットも存在するために，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．

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ので，これらに「BOT」,「Bot」,「bot」などの単語が含まれている場合は処理対象から除外する．

2.3 移動ツイートの抽出

ジオタグ付きツイートからユーザの位置は取得できても，その時に自動車，バス，電車，新幹線などの交通機関に乗車していたかどうかは判別できない．その代わりに，同一ユーザの連続する二つのツイートの投稿位置と経過時間から，それぞれのツイートをおこなった地点間の移動速度 v と移動距離 d を求めて，それが想定している移動手段の判定条件に合致した場合に，想定した交通機関で移動したと判定する．この処理で得られたツイートを**移動ツイート**と呼ぶ．ここで同一ユーザの連続するツイートを対として扱うのは，前後に交通機関で移動したツイートを抽出するためだけであり，その後の処理では再び独立したツイートとして扱う．

この移動手段の判定条件は，想定した交通機関の特徴によって異なる．例えば，電車やバス，市街地を自動車で移動する場合は比較的低速で移動するのに対して，新幹線や飛行機，高速道路を自動車で移動する場合は高速である．さらに，電車やバスは短距離移動に用いられるのに対して，新幹線や飛行機は長距離移動に用いられることが多い．そこで，想定する交通機関の種類に応じて移動速度 v と移動距離 d の範囲を設定する．

2.4 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** [4] を用いて，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 ρ，角度を θ 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，時系列的な連続性は無視し，距離的に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．この二つの移動ツイートは，必ずしも同一ユーザではなく，エッジも交通路上に乗っているとは限らないために，通常の Hough 変換と違って入力に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，ノイズが除去する．

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. 矩形領域内の任意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，矩形領域内のユーザ数がユーザ数が少ない領域は特定ユーザの自宅や職場である可能性が高いので，閾値 T_u を下回る場合には処理対象から除外する．

さらに矩形領域 (i,j) 内の分布が特定部分に偏っているかどうかを表す指標である集中度 $c_{i,j}$ を，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．

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

集中度は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性質を持つことか

ら，閾値 T_c を超える場合のみを対象とする．

2.5 同一経路と推定される近似直線のグループ化

上記の処理で独立した短い直線群が抽出されるが，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，さらに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 1 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 α，l' の角度を β，中線の角度を γ 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ために，矩形領域 (x, y) の近似直線 l の相手の近似直線 l' の探索範囲を $(x-2, y-2), (x+2, y-2), (x-2, y+2), (x+2, y+2)$ の 24 個の矩形領域とする．これにより，例えば隣接 8 矩形領域に近似直線が見つからない場合でも，その先の 16 矩形領域に近似直線が存在すればグループ化できる．

具体的なアルゴリズムを以下に示す．

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. $diff(\alpha, \beta) \leq T_{\theta_1}$ を満たさなければ，(1) に戻る．
3. $diff(\alpha, \gamma) \leq T_{\theta_2} \wedge diff(\beta, \gamma) \leq T_{\theta_2}$ 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 α，l' の角度を β，l と l' の中線の角度を γ，$diff(\theta, \theta')$ は，角度 θ，θ' の角度差 $\theta''(0 \leq \theta'' < 360)$ を求める関数，T_{θ_1}，T_{θ_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，近似直線グループサイズが閾値 T_g より小さい場合には，明確な交通路ではないノイズと考えて除外する．

3 評価

3.1 ツイートデータセット

Twitter Streaming API[1] 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，2012 年 5 月 1 日〜2014 年 4 月 31 日の 24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 306,694,602 である．

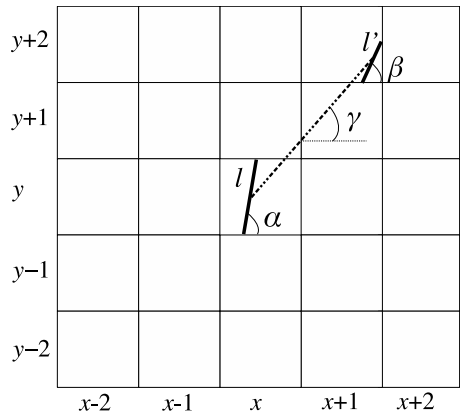

図 1 近似直線同一経路集合を構築する例

さらに，後述する理由から，Twitter クライアントを Twitter for iPhone と Twitter for Android に限定した．

3.2 パラメータ

交通路抽出を行った時のパラメータ設定は以下の通りである．

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，ユーザ数の閾値を T_u を 2，集中度の閾値 T_c を 0.3 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_{θ_1} と T_{θ_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．

3.3 移動速度・移動距離の分布

まず，最初に交通機関の特徴を把握するために，ユーザの連続する二つのツイートの投稿位置と経過時間から求めた移動速度 v と移動距離 d の分布を調べる．2013 年 5 月の 1ヶ月間の Twitter for iPhone 又は Twitter for Android を使用しているユーザに限定し，平均歩行速度 5km/h を上回る場合に何らかの交通手段を利用したと考えて，それらのツイート間の移動速度と移動距離をプロットした結果を図 2 に示す．横軸はツイート間の移動速度 (km/h)，縦軸は移動距離 (km) である．ただし，図 2 の横軸は 2000km/h までしかプロットしていないが，移動速度がそれを上回る場合があり，最大移動速度は約 800 万 km/h であった．

まず，移動速度が 5〜300km/h で移動距離が 0〜700km の領域に集中的に分布していることがわかる．東海道新幹線の最高速度が 285km/h，東京〜新大阪間の距離が 552.6km であることを考慮すると，この部分が電車，自

[1] https://dev.twitter.com/docs/api/streaming

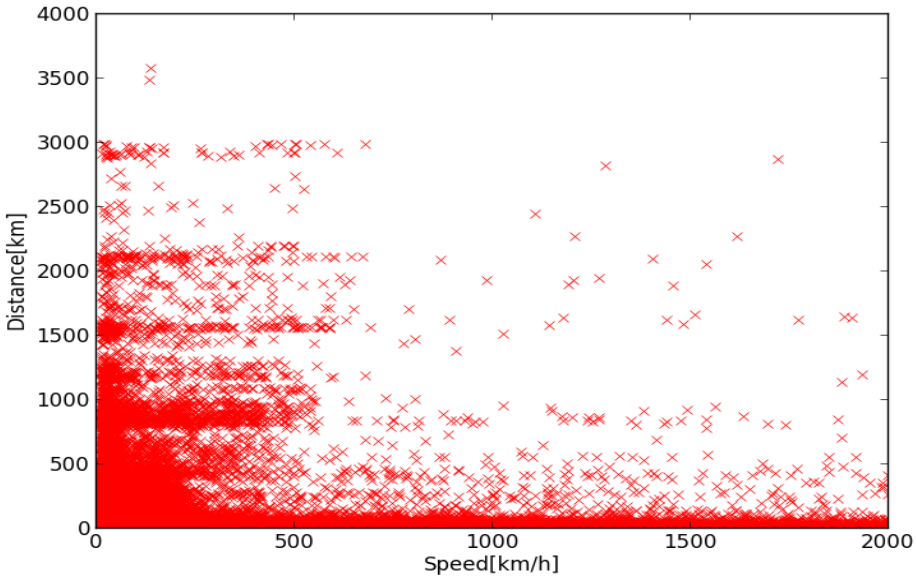

図 2 連続するツイート間の移動速度・移動距離の分布

動車，新幹線などの一般的な交通機関で移動した場合だと考えられる．

また，図 2 の左部分に注目すると，移動速度が 5〜700km/h で，移動距離が 800〜1000km，1500〜1600km，2000〜2100km，3000km 付近で飛び飛びに横長の分布が観測できる．羽田空港〜新千歳空港間の距離が 894km，羽田空港〜那覇空港間の距離が 1687km，成田空港〜北京空港が 2137km，成田空港〜香港空港が 2930km ということを考えると，この横長の分布は飛行機による移動だと考えられる．つまり，飛行機の飛行中は原則として機内における電波を発する電子機器の使用が禁止されているためにツイートすることができないことと，飛行機による移動距離と比較すると前後の移動距離は相対的に非常に小さくなることから，同じ路線ではほぼ同じ移動距離になるが，その前後のツイートまでの時間によって見かけの移動速度は大きく変化したように見えると推測する．

なお，移動距離が 0〜50km においては，移動速度が 5〜2000km/h まで幅広く分布している．しかし，旅客機の最大速度が 1030km/h であることを考慮すると，ユーザが実際にこのような速度で移動したとは考えられない．例えば，測位された位置情報に何らかの問題があり，誤った移動速度が計算されていると推測される．

以上の結果から，連続ツイート間の移動速度と距離を考慮すると，交通機関の種類ごとの経路抽出や，何らかの原因で発生したノイズを除去できる可能性があることがわかる．

3.4 異常な移動速度の原因の分析

次に，異常な移動速度を示した原因について分析した．

まず，異常な移動速度が計算されたクライアントにおいて，実際にどのような位置情報が付加されていたかを分析した．例えば，foursquare クライアントでは，「I'm at 地下鉄 名古屋駅 (Subway Nagoya Sta.)」など，訪れたいスポットに関しての情報を位置情報とともに発信していた．ただし，必ずしもツイート時にそのスポットに滞在しているとは限らず，実際の位置とは異なる位置情報を付加している場合があった．また，earthquake_JP7 クライアントでは，地震の発生を知らせるツイートに，その地震の震源地の位置情報を付加して発信していた．他にも同様に，そのアカウントのユーザの実際の位置ではない位置を付加してツイートするクライアントが存在した．

そこで，実際の位置情報を付加している可能性が高いクライアントとして，特に Twitter の公式アプリである Twitter for iPhone と Twitter for Android に注目し，ツイートに付加した位置情報について分析した．まず，10 日分のツイートデータに対して，これらのクライアントの使用率を調べると，iPhone が 29.2%，Android が 39.3%，合計 68.5%となり，この二つのクライアントのデータを使用するだけでも，充分なデータが得られることがわかった．ただし，図 2 で示したように通常考えられない移動速度である 1500〜2000km にも分布しているために，この二つのクライアントが付加する位置情報もすべて正しいわけではない．そこで，同じ 10 日分のツイートデータから Twitter for iPhone と Twitter for

表 1 交通手段の移動速度と移動距離の範囲

交通手段	移動速度 km/h	移動距離 km
低速・短距離交通手段	30〜70	5〜10
高速・長距離交通手段	70〜300	40〜300
飛行機	500〜800	800〜3000

Android による平均歩行速度 5km を上回るツイートを抜き出して二つのクライアントの比率を調べると，iPhone が 46.3%，Android が 53.7%であったが，1500〜2000km の範囲のツイートを調べると Android が 85.9%と非常に高い割合になった．この理由としては，Android は端末ごとに GPS 精度にばらつきがあることや，GPS 以外に Wi-Fi や携帯などの情報も用いて測位する場合に iOS の Core Location フレームワークよりも Android の Location API の方が測位精度が低いことが考えられる．

以上から，現在位置以外の位置を付与するクライアントが存在すること，比較的信頼できると思われる Twitter の公式クライアントの場合でも，Android においては取得する位置に大きな誤差があることがわかる．

3.5 移動速度・距離を考慮した交通路抽出の評価

最後に，移動速度・距離条件を変更し移動ツイートを抽出することで，実際に異なる交通手段の経路の抽出を行えるかを評価する．

これらのツイートの移動速度・距離帯の分布，交通手段の特性及び評定速度，そして測位誤差によるノイズの除外を考慮して，表 1 に示すように，低速・短距離交通手段，高速・長距離交通手段，飛行機の 3 種類に分析した．低速・短距離交通手段は，移動速度が遅く，停止頻度が比較的高い電車やバスなどを意図している．高速・長距離交通手段は，移動速度が速く，停止頻度が低い新幹線や急行などを意図している．飛行機は，飛行中は機内でツイートできないことから，これらとは別に扱う．なお，各移動速度と移動距離の範囲の設定においては，抽出精度は向上するように，明確に分離するためにギャップを設けている．このギャップにより抽出対象となるツイートは減少するが，そもそも扱うツイート数が多いために，大きな問題にはならない．

低速・短距離交通手段と高速・長距離交通手段に対して，山手線周辺区域を対象に抽出した交通路を Google Maps 上に可視化した結果を，それぞれ図 3 と図 4 に示す．

図 3 では，山手線周辺の電車路線が良好に抽出されている．ただし，国道や高速道路はうまく抽出されていない．その原因としては，自動車を運転中はツイートできないために，同乗者によるツイートのみに限られることから，必然的にツイート数が少なくなってしまうことが

図 3 低速・短距離交通手段を対象とした交通路の抽出

考えられる．また，図 4 では，池袋駅，新宿駅，渋谷駅周辺にノイズと思われる近似直線が抽出されているが，東海道新幹線と新幹線路線が比較的良好に抽出されている．

さらに，図 3 と図 4 のどちらにも，共通部分である田端駅〜東京駅〜品川駅の経路が抽出されており，異なる移動条件でもうまく抽出しわけていることがわかる．ただし，どちらの可視化結果にも，正確に抽出されていない部分やもしくは抽出できなかった路線があるが，それは同一矩形領域の複数路線の存在，正確な測位が d けいない地下部やトンネル部であることなどが原因であると推測している．

飛行機の飛行中は，原則として機内における電波を発する電子機器の使用が禁止されているためにツイートすることができないので，本手法では経路を抽出できない．しかし，今回指定した移動速度・移動距離の条件で抽出した移動ツイートの分布を知るために，抽出された関西地区周辺の移動ツイートを可視化した結果を図 5 に示す．この結果を見ると，関西地区の空港である伊丹空港，神戸空港，関西国際空港の 3 箇所に移動ツイートが集中していることがわかる．なお，それ以外の場所にもツイートが除去されずに残っているが，これは本手法を適用する前の段階のツイートをプロットしただけだから

図 4 高速・長距離交通手段を対象とした交通路の抽出

である．

4 おわりに

本稿では，入力とするツイート集合の移動速度と移動距離の条件を変更することで，異なる特性の交通手段の経路を抽出しわける手法を提案した．

まず，連続するツイート間の移動速度と移動距離の関係を分析し，移動速度と移動距離に着目することで移動手段がある程度推定できることと，クライアントによる任意の位置の指定や測位誤差の発生による異常な移動速度が発生することを示した．

さらに，移動速度と移動距離の分布から，交通手段を低速・短距離交通手段，高速・長距離交通手段，飛行機の 3 種類に分類して本手法を適用した結果を可視化し，それぞれ電車路線と思われる経路と新幹線と思われる別の経路が抽出できることと，飛行機を前提として抽出した移動ツイートが空港のある場所に集中することを示した．

今後の課題は，今回は移動手段の推定時に，ツイート間の移動速度と移動距離だけでは機械的にうまく分離できなかったことから，交通手段の特性及び評定速度を考えて値の範囲を設定したが，この処理の自動化と精度の向上が行える新たな特徴軸を設定することである．これ

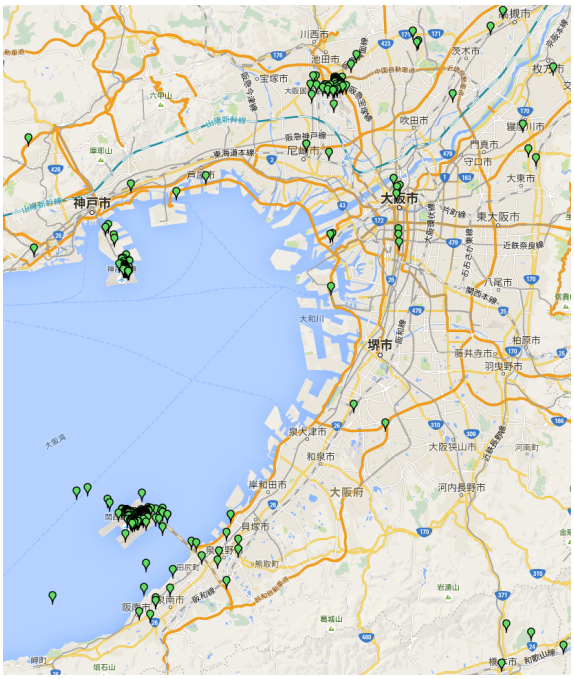

図 5 飛行機を対象とした移動ツイートの可視化

ができれば，例えば，低速・短距離交通手段において，さらに電車，バスを分離することができると考えられる．

謝辞

本研究は JSPS 科研費 26330345 の助成を受けた．

参考文献

[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] 谷直樹, 風間一洋, 榊剛史, 吉田光男. ジオタグ付きツイートを用いた交通路の抽出. *DEIM 2015 F7-4*, 2015.

[3] 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.

[4] 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.