

アドホック通信を用いた移動ノードの密度推定法

濱 田 淳 司[†] 内 山 彰^{†,††} 山 口 弘 純^{†,††}
楠 本 真 二[†] 東 野 輝 夫^{†,††}

本稿では、無線アドホック通信を用いて、携帯情報端末を保持する歩行者や車載端末を搭載した車両（ノード）の密度分布をリアルタイムに推定する方法を提案する。各ノードが周辺のノード密度分布を把握できれば、交通状況に応じた高度ナビゲーションや混雑するイベントでの歩行者誘導など高度交通システムにおける様々なサービスなどへの応用が期待できる。提案手法では、各ノードはGPSなどで自身のおおよその位置を把握できるものとし、隣接端末が保持するノード分布情報をアドホック通信で定期的に受信することで自身が把握するノード分布情報を更新する。また各ノードがノード分布の変化予測を行うことで時間経過によるノード分布の変化にも追従する。シミュレーション実験を行い、実密度分布と推定密度分布に対して単位領域ごとの密度値の順位付けの相関を導出した結果、相関係数が0.64から0.84となり、両者の相似度が十分高いことが示された。

Density estimation for mobile nodes using ad hoc communication

JUNJI HAMADA,[†] AKIRA UCHIYAMA,^{†,††} HIROZUMI YAMAGUCHI,^{†,††}
SHINJI KUSUMOTO[†] and TERUO HIGASHINO^{†,††}

In this study, we propose a method for mobile wireless nodes, which may be pedestrians or vehicles with information terminals, to estimate the density of mobile nodes in their surroundings. The method enables to provision intelligent services which are environment-aware with highly dynamic movement of nodes, like intellectual navigation that tells the user the best route to detour congested region. In the proposed method, each node is assumed to know its location roughly (*i.e.* within some error range) and to maintain a density map covering its surroundings. This map is updated when a node receives a density map from a neighboring node. Also by estimating the change of the density, taking into account the movement characteristics of nodes, it is updated in a timely fashion. The simulation experiments have been conducted and the correlation between the ranks of density values of unit cells in the real and estimated density maps has been measured. The results in two different scenarios have shown that the proposed method could attain the correlation coefficients 0.64 and 0.84, indicating the high accuracy of the estimated density maps.

1. ま え が き

近年、無線通信技術の発達や携帯端末の普及により、端末間のアドホック通信を用いた新しいネットワークアプリケーションが注目を集めている。例えば高度交通システムにおいては、車両間通信（Inter-vehicle communication, IVC）を用いた事故防止システムが実用化に向けて研究開発が進められている [1] ほか、Vehicular Ad-hoc Network (VANET) を介した交通情報ならびに車両情報の収集や散布などについて多くの研究がなされている [2–5]。

それらの研究事例では、アドホック通信を用いることでこれまでインフラストラクチャに依存してきたデータ収集やデータ配信を低コストかつ高効率で行えることを示唆している。なかでも、移動する人や車が、同じく周辺に存在する人や車の分布状況をリアルタイムで取得することができれば、多く

のアプリケーションならびにサービスにおいて極めて有用であると考えられる。例えば、避難誘導や混雑回避などを目的としたナビゲーションにおいても、周辺領域の密度分布が分かれば安全性や効率を考慮したサービスが可能となる。しかし、移動端末密度分布の推定を端末協調で効率よく行うアプローチは現状ではまだ十分に研究がなされていない。

そこで本研究では、各移動端末が周辺の移動端末の密度分布をリアルタイムに推定する方法を提案する。提案手法ではGPSなどにより各端末が自身の位置情報を（ある一定の誤差範囲で）取得できると仮定する。各端末は、自身が知り得る周辺領域の密度分布情報に自身の位置情報を重畳し、周辺端末にアドホック通信でその密度分布情報を伝搬させる。これを繰り返す行うことで、各端末は対象領域の密度分布を得ることができる。

以降では、移動端末の密度分布情報を密度マップとよぶ。本研究の目標は、できる限り多くの端末に、なるべく広範囲かつ正確な密度マップを自律的かつ低トラフィックで構築させることにある。領域全体の密度マップを構成するため、各端末は適当なタイミングで、自身が存在する可能性のある領域情報を加えた密度マップを定期的に隣接端末にブロードキャ

[†] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

^{††} 独立行政法人科学技術振興機構, CREST
Japan Science Technology and Agency, CREST

ストし、隣接端末がブロードキャストした密度マップを受信した場合は自身の密度マップを更新する。なお、この方法では明らかに情報の新鮮さと伝搬距離に応じてトラフィックが増大するが、提案手法では密度マップの正確性を維持しつつ通信量を抑制するため、端末密度が端末の移動にしたがってどのように変化するかを予測する関数（以降、予測関数）を導入する。予測関数はあらかじめ与えるものとし、各端末は時間経過にしたがって予測関数を用いて自身の密度マップを更新する。対象領域における端末の移動特性が分かる場合には、予測関数にそれらの情報を反映させることができる。例えば端末の最大移動速度 V_{max} のみが分かる場合に、経過時間 Δt として密度マップの各地点の密度を $V_{max} \cdot \Delta t$ だけ全方位に拡散させたり、端末が自動車の場合には、同様に各地点の密度を拡散させるが、拡散方向は道路（密度が極端に高い領域）上のみに制限するといった予測関数が考えられる。また、通信量を抑制するため、周辺端末密度に応じてメッセージ送出頻度を調整する方法も提案している。

そしてシミュレーション実験を行い、実密度分布と推定密度分布に対して単位領域ごとの密度値の順位付けの相関を導出した結果、相関係数が 0.64 から 0.84 となり、両者の相似度が十分高いことが示された。

2. 関連研究

VANET においては、高度交通システムで用いられるコンテキスト（路面情報や気象情報、交通情報等のデータ）の収集ならびに配信を念頭においた様々な手法が提案されている [2–5]。また、無線センサネットワーク（Wireless Sensor Network, WSN）においては、例えば地理的に近い場所で観測された気温観測値など類似性が高いために冗長度が高いデータなどを集約しネットワークトラフィックを削減する方法や、与えられた query に対し、WSN ノードがそれぞれ中間状態を計算する方法など、多くのデータ集約手法が提案されている [6–9]。

提案手法は各ノードのセンシングデータをなるべく少ないトラフィックで集約する点で上記のアプローチと共通する。しかし、提案手法ではモバイルノードの位置情報といった分単位のレベルで変化する情報を扱っており、WSN や VANET で収集する気象情報、路面状況や広域交通情報など、比較的長時間に渡り安定的なデータとは異なる。一方でプロトコルのオーバーヘッドは少ない方が望ましいため、それらのトレードオフを考慮しながら適切な手法を設計する必要がある。また、各ノードが頻繁に移動する動的な環境で動作する必要もある。これらの点で扱う問題は大きく異なる。本手法ではこの問題に対し、各ノードが現在の密度分布の時間変化をノードの移動特性を考慮して予測し、さらに密度に応じた送信頻度を設定することで、センシングデータ交換にかかるトラフィックを抑制する自律分散プロトコルを提案しており、高い新規性を備えた合理的なアプローチであるといえる。

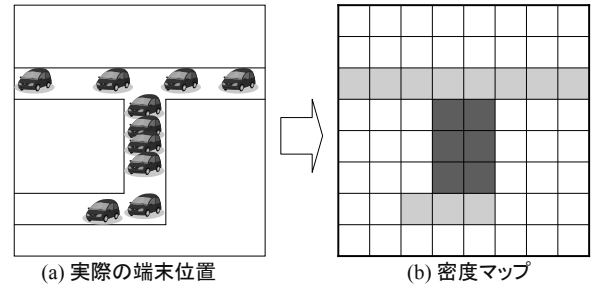


図 1 密度マップの例

表 1 プロトコルパラメータ

記号	意味
s	正方セルの辺長 (m)
$f(D)$	密度マップ時間変化の) 予測関数
V_{max}	端末の最大移動速度 (m/s)
t	密度マップ送信間隔 (s)
TH_d	有効最小密度 (ノード/セル)
ε	単一ノードの有効最小密度 (ノード/セル)

3. 密度分布推定法

3.1 概要

図 1 に密度マップの例を示す。また、以降の説明で用いるプロトコルパラメータを表 1 に示す。領域は $s \times s$ (m) の正方形（セルとよぶ）に分割されているものとし、各端末 i は自身の位置を中心とした密度マップ D_i を保持する。 D_i は $X \times Y$ 個の要素を持ち、各要素 $d_{x,y}$ ($1 \leq x \leq X, 1 \leq y \leq Y$) はセル (x, y) における密度を表す。本稿では説明の簡単のため各端末の密度マップの範囲は同じものとするが、アルゴリズムは各端末が設定する密度マップの大きさには依存しない。また、各端末は全端末の最大移動速度 V_{max} (m/s) を知っているものとする。これは車両であれば最高制限速度などから推測可能である。

各端末 i は以下の処理を密度マップ送信時間間隔 t ごとに行う。

(1) 密度マップ D_i を予測関数 $f(D_i)$ によって更新する。予測関数には対象環境において一般的な移動パターンがモデル化されて与えられるとする。このモデルに従って、各セル内のノードが移動したとみなし、予測関数により各セルの密度値を、時間 t 内で移動可能なセルに配分する。更新後のセル値 $d_{x,y}$ が有効最小密度 TH_d 以下となった場合には、そのセルには端末が存在しないと見なし、 $d_{x,y}=0$ にリセットする。 TH_d には密度データとして有効な最小の値が設定されているものとする。

(2) 後述する条件を満たす場合、自身の位置情報 R_i を自身の密度マップ D_i に登録する。端末 i の位置情報 R_i は、 n 個のセルを要素に持つ集合

$$R_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \dots, (x_{in}, y_{in})\}$$

により領域として表される。位置情報の登録が可能な条件は、端末 i が最後に位置情報の登録を行ってからの経過時間がある Δt_i (秒) 以上の場合のみとする。 Δt_i には最後に登録し

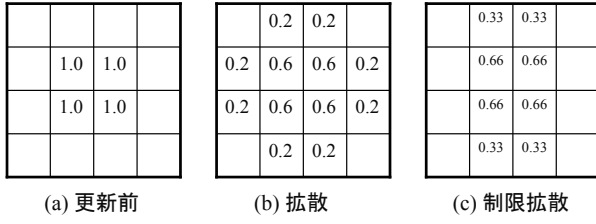


図 2 予測関数の適用例

```

for(step=0; step < [t * Vmax/s]; step++){
  D'i ← Di;
  foreach (dx,y ∈ Di){
    d'x,y ← dx,y + 0.2 * dx,y;
    d'x-1,y ← dx-1,y + 0.2 * dx,y;
    d'x,y-1 ← dx,y-1 + 0.2 * dx,y;
    d'x+1,y ← dx+1,y-1 + 0.2 * dx,y;
    d'x,y+1 ← dx-1,y-1 + 0.2 * dx,y;
  }
  Di ← D'i;
}
return Di;

```

図 3 拡散予測関数

た位置情報 R_i の密度情報が予測関数により拡散していき、単一ノードの密度情報として有効な最小の値（単一ノードの有効最小密度） ε 以下になるまでの時間を設定する。従って、予測関数により Δt_i は異なる。

- (3) 密度マップ D_i を隣接端末へブロードキャストする。
- (4) 他端末 j から密度マップ D_j を受信した場合には D_i を更新する。

以下 3.2 節でこれらの手続きを説明する。

3.2 アルゴリズム

3.2.1 予測関数

密度マップはあらかじめ与えられた予測関数 $f(D)$ によって更新される。予測関数には対象領域/対象端末について特徴的な移動パターンを組み込む。一般的な移動パターンとその予測関数の例として、ここでは (i) 最大移動速度 V_{max} 以外に移動パターンが不明な場合（拡散予測関数）(ii) 移動可能領域が決まっている場合（制限拡散予測関数）、および (iii) 自身の周辺は制限拡散予測関数を適用し、それ以外は拡散予測関数を適用する場合（ハイブリッド予測関数）について述べる。

拡散予測関数 最大移動速度 V_{max} のみが分かっている場合、各端末は領域内をあらゆる方向に移動する可能性がある。従って拡散予測関数では、各セル値を上下左右のセルに均等配分する。セルの大きさは $s(m)$ であり、予測関数による更新は t 秒ごとに行われるので、この処理を $\lfloor t * V_{max}/s \rfloor$ 回繰り返す。この予測関数 $f(D_i)$ の適用例と擬似コードをそれぞれ図 2(b)、図 3 に示す。この予測関数において、次に自身の位置情報 R_i を D_i に登録するまでの時間 Δt_i (秒) は、以下の条件を満たす k に基づき決定する。

$$\frac{1}{|R_i|} * \left(\frac{1}{5}\right)^k \leq \varepsilon \quad (1)$$

```

for(step=0; step < [t * Vmax/s]; step++){
  D'i ← Di;
  foreach(dx,y ∈ Di){
    expand_# ← 1;
    sum ← 0;
    for(j ← 1; j ≤ n; j++){ sum += dx+j,y;
    avg ← sum/n; right ← false;
    if(avg ≥ THmove){
      right ← true; expand_#++;
    }
    sum ← 0;
    for(j ← 1; j ≤ n; j++){ sum += dx-j,y;
    avg ← sum/n; left ← false;
    if(avg ≥ THmove){
      left ← true; expand_#++;
    }
    sum ← 0;
    for(j ← 1; j ≤ n; j++){ sum += dx,y+j;
    avg ← sum/n; down = false;
    if(avg ≥ THmove){
      down ← true; expand_#++;
    }
    sum ← 0;
    for(j ← 1; j ≤ n; j++){ sum += dx,y-j;
    avg ← sum/n; up ← false;
    if(avg ≥ THmove){
      up ← true; expand_#++;
    }
    if(right) d'x+1,y ← dx+1,y + 1/expand_# * dx,y;
    if(left) d'x-1,y ← dx-1,y + 1/expand_# * dx,y;
    if(down) d'x,y+1 ← dx,y+1 + 1/expand_# * dx,y;
    if(up) d'x,y-1 ← dx,y-1 + 1/expand_# * dx,y;
    d'x,y ← dx,y + 1/expand_# * dx,y;
  }
  Di ← D'i;
}
return Di;

```

図 4 制限拡散予測関数

ここで、 k は予測関数によるセル値の拡散回数である。セル値の配分処理は s/V_{max} 秒あたりに 1 回適用されるので、

$$\Delta t_i = \frac{k * s}{V_{max}} \quad (2)$$

となる。

移動可能領域を考慮した予測関数 移動可能領域が決まっている場合の例として、道路で人や車の移動が挙げられる。ここでは特別な地図情報を仮定せずに、密度マップから得られる情報を利用して移動可能領域を推測し、その移動可能領域のみに各セルの値（密度）を分配するための予測関数（制限拡散予測関数）を考える。

この予測関数の適用例および擬似コードをそれぞれ図 2(c)、図 4 に示す。この予測関数では各セル $d_{x,y}$ から上下左右それぞれについて、 m グリッド先までの平均密度がある閾値 TH_{move} 以上であれば、その方向は移動可能領域であると見なし均等に $d_{x,y}$ の値を配分する。この処理は $\lfloor t * V_{max}/s \rfloor$ 回繰り返される。

制限拡散予測関数の場合、更新するごとにセルの値を均等に分割する個数が密度によって変わるため、登録した位置情報の影響が消失するまでの時間 Δt_i を求めることは複雑である。従って、簡単のため拡散予測関数の場合と同じ条件に基づき Δt_i を決定する。

ハイブリッド 密度マップはノード間で送受信されること

で徐々に伝搬していくため、現在位置から遠いほど情報の鮮度は落ちる．このため移動可能領域の推測がうまく行えない可能性がある．そこで、自身の現在位置周辺のセルに対しては制限拡散予測関数を適用し、それ以外のセルに対しては拡散予測関数を適用する．

ここでは、自身の現在位置周辺を位置情報 R_i に含まれるセルと定義する．密度マップ D_i について、 R_i に含まれるセル $d_{x,y}$ に対しては拡散予測関数を用い、それ以外のセルに対しては、移動可能領域を推測し、拡散方向を制限する予測関数を用いる．次に位置情報を登録するまでの時間 Δt_i は、簡単化のため拡散予測関数と同じ方法により決定する．

3.2.2 位置情報の登録

登録後の密度マップを D'_i とすると、その各要素 $d'_{x,y}$ は以下の式により決定される．

$$d'_{x,y} = \begin{cases} d_{x,y} + \frac{1}{n}, & \text{if } (x,y) \in R_i; \\ d_{x,y}, & \text{otherwise.} \end{cases} \quad (3)$$

ここで、 n は R_i の要素数である．式 (3) では位置情報 R_i の精度が低い、すなわち R_i の面積が大きいかほど R_i に含まれるグリッドに加算される値が小さくなる．

3.2.3 密度マップの併合

端末 i が密度マップ D_j を受信した場合には D_i を更新する．密度マップには各セル値の新しさを示す時刻情報が付与とされていないため提案手法では D_i と D_j の各セル (x,y) の密度情報の新しさを $d_{x,y}$ の大きさに判定する．これは、各セルの密度は時間経過とともに分散するため、セルの値が大きいかほど新しい情報である可能性が高いという観測に基づいている．そこで密度マップの更新では、各セル (x,y) について、以下のように2つのマップのうち値の大きい方を更新後の密度マップのセル値 $d'_{x,y}$ とする．

$$d'_{x,y} = \max\{d_{x,y}^{(i)}, d_{x,y}^{(j)}\} \quad (4)$$

3.3 通信量の削減

各端末 i は t 秒ごとに密度マップ D_i を送信する． D_i はセルにより表されるためそのデータサイズはセル面積 s^2 に反比例し、領域の面積に比例する．提案手法では、通信量と精度のトレードオフを追求するため、隣接端末数に応じて送信する密度マップの範囲を調整する方式を導入する．

以下、密度マップ D_i の部分マップを $\hat{D}_i$ で表し、サブ密度マップとよぶ．密度情報を遠方に伝搬させ、各端末の密度マップの精度を向上させるためには、全端末が密度マップ全体を常に配信することが望ましい．しかし、付近に多数の端末が存在する場合、それらの端末の保持する密度マップは遠方のデータの類似度が高いと想定されるため、周辺密度に応じて送信する密度マップの範囲を調整し、密度に依存せず常にいくつかの端末のみが密度マップを送信すれば十分である．

このアイデアに基づき、提案手法では各端末 i は隣接端末数 N_i に反比例するよう

$$S_i = \frac{\alpha}{N_i} \quad (\alpha \text{ はある定数}) \quad (5)$$

のように定義した S_i をその大きさとするサブ密度マップ $\hat{D}_i$

表 2 シミュレーション諸元

パラメータ	設定値
領域サイズ (m)	100 × 100
境界遭遇時	跳ね返り
交差点における方向選択	ランダム
道路幅 (m)	20
セルの辺長 s (m)	2
無線通信距離 R (m)	10
端末の移動速度 (m/s)	0.1 ~ 1.0 $V_{max} = 1.0$
シミュレーション時間 (s)	600
帯域幅 (Mbps)	1
端末数	200
予測関数 $f(D)$	ハイブリッド予測関数 ($m = 10$)
有効最小密度 (ノード/セル) TH_d	0.005
単一ノードの有効最小密度 (ノード/セル) ε	0.0035
R_i の平均サイズ (m ²)	cross:349.08 Manhattan:238.28
サブ密度マップの送信間隔 t (s)	2
密度マップの最大送信間隔 T (s)	10

を用い、 t 秒毎に $1/N_i$ の確率で密度マップ D_i を送信し、そうでない場合はサブ密度マップ $\hat{D}_i$ を送信する．さらに、一定時間内に必ず領域全体の密度マップが送信されることを保証するため、ある T に対し、過去 T 秒間に一度も密度マップを送信していない場合は、必ず密度マップをブロードキャストする．これらにより、冗長な送信データを削減できる．

4. シミュレーション実験

4.1 設定

ネットワークシミュレータ MobiREAL [10] を利用して提案手法の性能評価を行った．シミュレーション領域は 100m × 100m の正方領域とした．この領域の中心から東西南北に延びる幅 20m の直線道路上はノードの存在可能領域、それ以外の領域は存在不能領域とする．道路モデルとして東西、南北それぞれ1本ずつの道路からなる cross-road モデル (図 5(a))、およびそれぞれ2本の道路からなる Manhattan-road (図 5(b)) モデルの2通りを用い、初期状態では端末をランダムに配置した．各端末は道路の交差点では4方向のいずれかを等確率で選択し、領域境界に達した場合は反対方向に移動方向を変更するモビリティモデルにしたがい、等速で移動する．端末の移動速度は、端末ごとに 0.1m/s ~ 1.0m/s の間でランダムに決定し、600 秒のシミュレーションを行った．シミュレーション諸元を表 2 に示す．またシミュレーションの結果、密度マップの精度は全端末でほぼ同程度であったため、以降のシミュレーション結果では特に指定がなければ密度マップは実験で着目したある端末 p の密度マップを指す．端末 p は cross-road (図 5(a)) や Manhattan-road (図 5(b)) を図 6 に示すように移動する．

4.2 評価結果

4.2.1 密度マップの推定総端末数

2つの道路モデルのシミュレーションを行った結果の実際

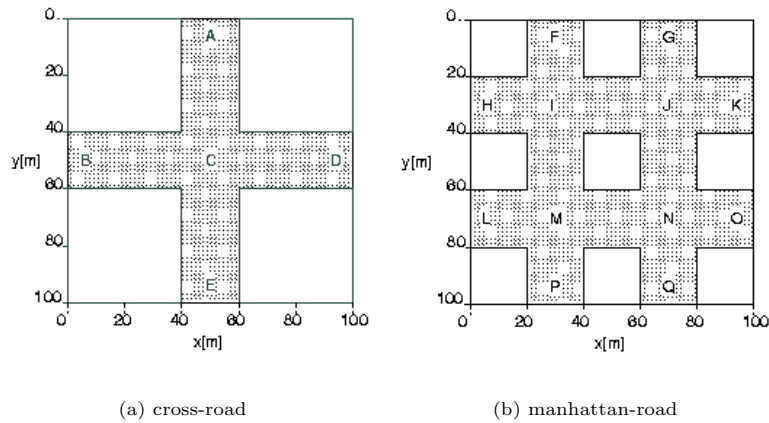


図 5 道路モデル

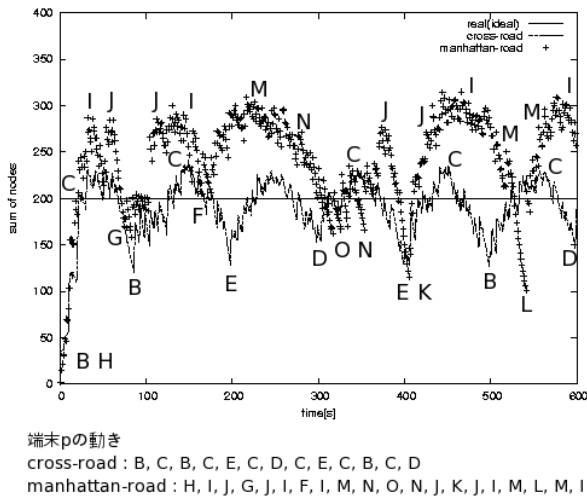


図 6 端末 p の密度マップの推定総端末数

の端末分布及び端末 p の密度マップの推定総端末数を図 6 に示す。2 つの道路モデルにおいて、ともに 0s ~ 20s の間は密度マップの推定密度と実際の端末分布の誤差が大きいが、これは密度マップに自身の周辺の密度分布しか反映されていないからである。密度マップの送受信によって広範囲の密度分布情報を取得することで、20s 時点までに密度マップの推定総端末数は急激に増加し、20s 以降は周期的に増減を繰り返している。

cross-road の場合では、端末の存在する位置によって密度マップの端末総数が最大 50 程度異なるが、端末が保持する密度マップにおける 21s ~ 600s までの推定総端末数の平均値は 193.00 であり、実際の総端末数 200 に極めて近い値となっている。cross-road では 2 本の道路が交差しており、端末が存在不可能な 30m × 30m の部分領域が 4 つある。密度マップに予測関数を適用することによって密度分布は広がるが、有効最小密度以下のセルの密度をリセットすることによって、端末が存在不可能な領域の中で、道路から離れているセルの

密度は 0 になるため、密度誤差が小さく抑えられている。端末が中心の交差点付近 (C) に移動してくると密度マップの推定総端末数は大きくなっている。これは、東 (B) 西 (D) 南 (E) 北 (A) の各方向の密度分布が取得できるからと考えられる。逆に中心の交差点 (C) から離れていくにつれて、密度マップの推定総端末数は小さくなっている。この理由は、中心の交差点 (C) から見て現在端末が存在する方向以外の領域から離れることにより、それらの領域の密度分布が取得しにくくなっているためと考えられる。

図 7(a) と図 7(b) は、450s 時点での cross-road マップにおける実際の密度分布及び端末 p が保持する密度マップである。450s 時点で端末 p は C 付近に存在している。密度マップと実際の密度分布形状を比較した場合、C からの距離が離れている A, B, E 付近では密度分布に少し誤差があるが、4 つの存在不可能領域の密度は 0 に近く、かつ C 付近の密度はともに高くなっている。

一方、Manhattan-road に着目して図 6 を見ると、cross-road と同様に端末が交差点付近 (I, J, M, N) に近付くと密度マップの総端末数は大きくなり、交差点から離れていくにつれて、密度マップの総端末数は小さくなるのが分かる。マップの推定総端末数は端末の存在する位置によって最大 100 程度の誤差があり、端末の保持する密度マップの 21s ~ 600s までの平均値は 245.64 と cross-road の場合よりも誤差が大きい。Manhattan-road では、4 本の道路により区切られた端末が存在不可能な領域が 8 つあるが、1 つの領域は 20m × 20m と cross-road のそれよりも小さい。そのため、予測関数により各道路上から拡散された値が複数重なることが多くなり、有効最小密度を下回るまでにかかる時間が cross-road の場合よりも長くなる。このような理由から、Manhattan-road では密度誤差の幅が大きくなっていると考えられる。

また、図 7(c) と図 7(d) は、450s 時点での Manhattan-road マップにおける実際の密度分布及び端末 p が保持する密度マップのスナップショットである。密度マップと実際の端末分布の密度分布形状を比較した場合、道路領域の密度形状

は似ているが、道路領域以外のエリアでは、実際の端末分布では密度は 0 になっているのに対し密度マップの場合は 0 以上のセルが見られ、密度に誤差があることがうかがえる。

4.2.2 密度分布の相似度

端末が保持する密度マップが、実際の密度分布に近いほどリアルタイムな密度分布状況を取得できているといえる。端末が保持する密度マップの密度分布と実際の密度分布の相似度を求めるために、ケンドールの順位相関係数を用いた。

cross-road の場合の順位相関係数を図 8(a) に、Manhattan-road の場合の順位相関係数を図 8(b) に示す。なお、セルサイズが密度マップと実際の密度分布の順位相関に与える影響を調べるために、順位相関係数を求める単位セルグループのサイズを 1×1 セル ($2m \times 2m$)、 2×2 セル ($4m \times 4m$)、 5×5 セル ($10m \times 10m$)、 10×10 セル ($20m \times 20m$) 場合の評価を行った。

図 8 より、2 つの道路モデルではともにシミュレーション開始後 20s 間は順位相関係数が急激に増加している。これは端末が密度マップの送受信によって、広範囲の密度分布状況を取得しているからである。広範囲の密度分布状況を取得後、順位相関係数は比較的安定する。単位セルグループのサイズが小さいほど、密度分布の誤差が順位相関係数に顕著に現れることは自明であるが、評価結果からはその様子が観測できる。なお、21s ~ 600s 間の順位相関係数の平均値は単位セルグループのサイズが 10×10 セルの場合、cross-road では 0.802 となりかなり強い相関が観測される。Manhattan-road の場合においても平均値は 0.644 となり十分な相関があるといえる。

4.3 通信量の削減

4.3.1 通信量の削減効果

提案手法では、密度マップをグリッドで管理している。シミュレーション実験において用いた密度マップは $100m \times 100m$ の領域であるため、1 グリッドのサイズが $2m \times 2m$ の場合、2500 個のグリッドから構成される。各グリッドの保持する値は、4B の実数型とすると、2500 個のグリッドで構成される密度マップのサイズは、圧縮をしない単純なデータ構造であれば 10KB になる。

本手法では、密度マップの作成・維持のために密度マップを送受信する。600s のシミュレーションにおいて、200 個の端末が 2s ごとに 10KB の密度マップを送信する場合、1 端末あたりの平均使用帯域幅は 40kbps に及ぶ。しかし前述のように、提案手法では各端末は 2s ごとにサブ密度マップが密度マップをランダムに送信し、10s 間密度マップが送信されなかった場合には強制的に密度マップを送信するようにして、通信量を削減している。通信量の削減前後のトラフィックサイズを表 3 に示す。通信量削減の工夫の結果、1 端末あたりの平均スループットは cross-road の場合 13.4kbps に、Manhattan-road の場合は 17.32kbps になった。これは無圧縮の密度マップ送信にかかるトラフィックであるため、密度マップの符号化などを行えば十分少ないトラフィックに抑えられると考えられる。

4.3.2 通信量削減と精度の関係

提案手法では密度マップの代わりにサブ密度マップを送信することで通信量を削減している。ここでは密度マップの送信頻度が密度マップの精度に与える影響を解析する。表 4 では、通信量の削減ありとなしの場合で、密度マップと実際の密度分布の順位相関係数の平均値を比較している。この結果より、2 つの道路モデルにおいて、通信量なしの場合が通信量の削減ありの場合より順位相関係数の平均値が高くなっていることが分かる。つまり、密度マップの送信を頻繁に行うほど、密度マップと実際の密度分布の相似度は増加する。通信量の削減と密度分布の相似度はトレードオフの関係にあり、環境に応じて適切に設定することが重要である。

5. 議 論

5.1 密度マップのデータ構造

提案手法では密度マップの表現に正方形セルを用いている。セルによるデータ構造はその取扱いや移動予測計算が簡易になる反面、セルサイズによってはデータ容量が増大する。WSN では同じデータをセンシングしたノードを結ぶ等高線を構成し、センシングデータ量を削減したり適切な query 送信に利用したりする手法 [6, 11] が知られているため、それらを用いたり、符号化によるデータ圧縮などを行うことも考えられる。いずれの手法もノードにおける計算量を増加させるため、通信コストと計算コストのトレードオフになるといえる。

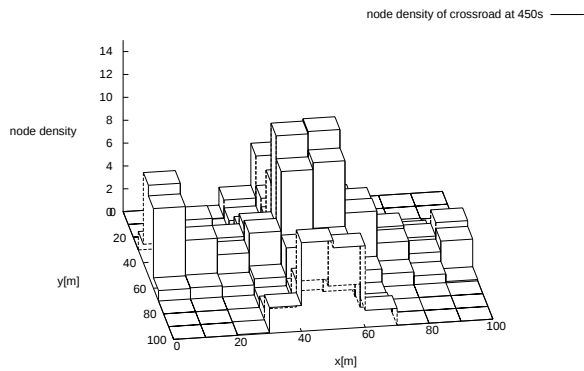
5.2 提供する位置情報の精度

提案手法では自身の位置情報を密度マップに付加する際、位置情報を点座標ではなく領域で指定できる。これにより、GPS などで生じる一定の位置誤差を吸収するのみならず、正確な位置情報を把握している場合も自身のプライバシー情報保護を目的とし意図的に領域情報を利用することもできる。また、GPS が利用できない場合でも UPL [12] のように移動端末が存在する可能性のある領域をアドホックネットワークを介して推定する手法などを用いることも可能である。

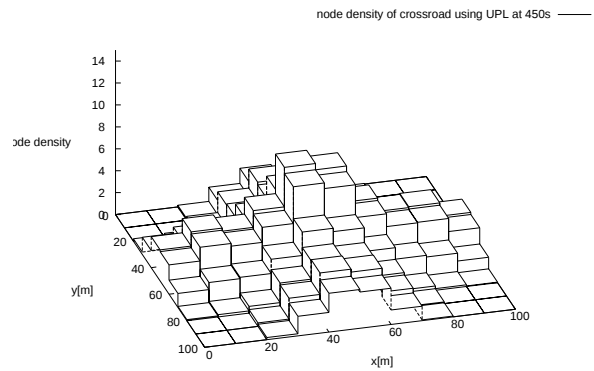
6. あとがき

本稿では、無線アドホック通信を用いて、ノードの密度分布をリアルタイムに推定する方法を提案した。シミュレーション実験を行い、実密度分布と推定密度分布に対し、単位領域ごとの密度値の順位付けの相関を導出した結果、相関係数が 0.64 から 0.84 となり、両者の相似度が十分高いことが示された。

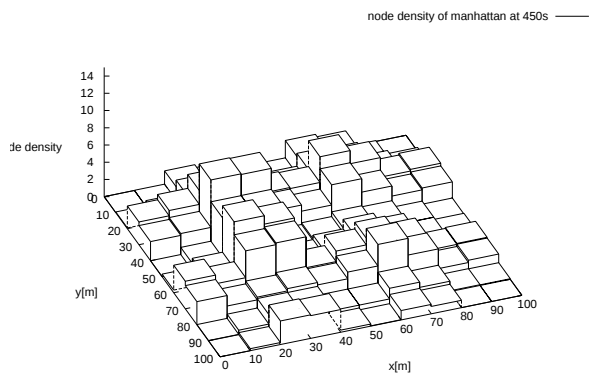
VICS などで提供されている道路交通状況では、車両感知器などで検出した車両情報を用いているため、基本的にはそれらが設置された国道などの主要道路が対象となっているが、提案手法により各車両もしくは路側機が周囲一定領域の車両の密度をリアルタイムに検出することで、インフラではカバーできていない局所的な道路状況を各車両が把握できると考えられる。周辺車両状況の把握により、インテリジェントなナビゲーションなど多くの応用が考えられる。また、近年プローブカーで路面状況や渋滞情報、天候情報などを収集する研究



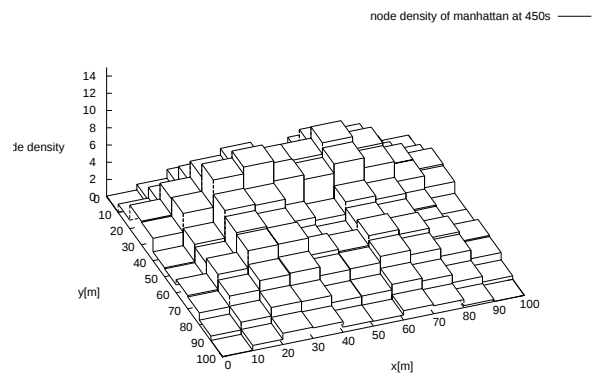
(a) 実際の密度分布 (cross-road)



(b) 端末 p の密度マップ (cross-road)

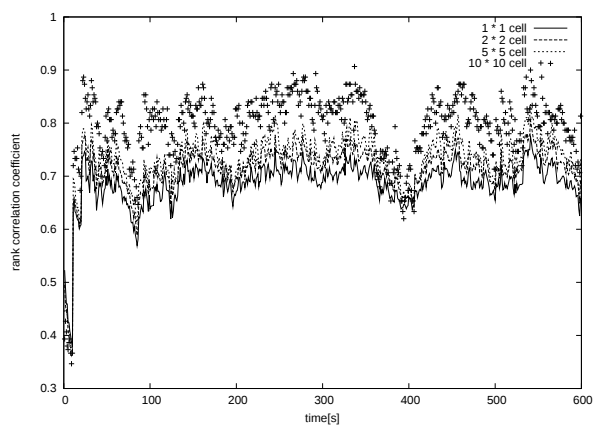


(c) 実際の密度分布 (manhattan-road)

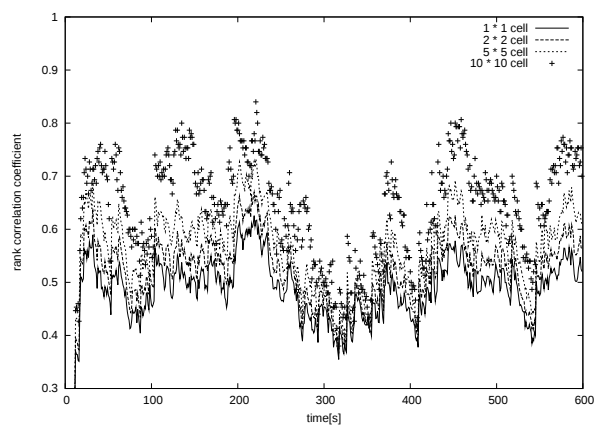


(d) 端末 p の密度マップ (manhattan-road)

図 7 実際の密度分布及び端末 p の密度マップ (450s 時点)



(a) cross-road



(b) manhattan-road

図 8 順位相関係数の時間変動

表 3 通信量削減の効果

	cross-road		Manhattan	
	削減なし	削減あり	削減なし	削減あり
密度マップ合計送信回数 (600s)	60000 回	13572 回	60000 回	14613 回
サブ密度マップ合計送信回数 (600s)	0 回	46428 回	0 回	45387 回
1 ノード当りの密度マップ平均送信回数 (600s)	300 回	67.86 回	300 回	73.07 回
1 ノード当りのサブ密度マップ平均送信回数 (600s)	0 回	232.14 回	0 回	226.9 回
合計通信量 (200nodes)	600MB	202.41MB	600MB	259.81MB
密度マップ合計通信量 (200nodes)	600MB	66.69MB	600MB	146.13MB
サブ密度マップ合計通信量 (200nodes)	0KB	2.67MB	0KB	113.68MB
1 送信当たりのサブ密度マップの平均サイズ (200nodes)	0KB	1.44KB	0KB	2.50KB
平均使用帯域幅 (1s, 1node 当たり)	40kbps	13.4Kbps	40kbps	17.32kbps

表 4 通信量の削減が平均順位相関係数に与える影響 (21s ~ 600s 間)

	順位相関用セルサイズ			
	1×1	2×2	5×5	10×10
密度マップを 2s 毎に送信 (cross-road)	0.715	0.749	0.777	0.839
サブ密度マップを 2s 毎に送信かつ密度マップを 10s 毎に送信 (cross-road)	0.691	0.719	0.738	0.802
密度マップを 2s 毎に送信 (Manhattan-road)	0.552	0.595	0.650	0.748
サブ密度マップを 2s 毎に送信かつ密度マップを 10s 毎に送信 (Manhattan-road)	0.491	0.526	0.569	0.644

や実証実験が行われているが、これらにおいて車載機が収集可能な情報量を増大させるなどの応用事例も考えられ、高度交通システムにおいて多様な状況における既存システムの補完的な利用事例が見込まれる。一方、大規模なショッピングモールや、花火大会など大規模な人員が集まるイベントなどでは、参加者が保持する携帯端末を介して群集の位置や行動を把握することで、各個人は適切な行動判断が、運営者側では群集の安全な誘導やマーケティングが可能となる。また、VANET や MANET において多くの位置情報ルーティングの研究がなされており、GPSR のように局所情報から障害物もしくは低密度地域を迂回する方法も考案されている。一方、提案手法のように低トラフィックでノード密度分布を把握できれば、それを元に各ノードがより広域でのメッセージ経路を決定し、それに基づくメッセージフォワーディングを行うなど、より効率のよいデータ配信手法を検討することも可能である。

今後は上記のような事例を対象として、より現実環境に即したシミュレーション実験を行い、各シナリオのもとでの課題発見やパラメータ設定、妥当性ならびに有用性検証などに取り組んでいく予定である。また、メッセージの送出タイミングなどプロトコルパラメータや対象領域範囲が周辺ノードの密度に応じて自律的に収束する枠組みを検討し、アドミニストレーションフリーに近いプロトコルを実現することが今後の目標である。

参 考 文 献

- 1) 国土交通省他: 先進安全自動車 (ASV) プロジェクト.
<http://www.mlit.go.jp/jidosha/anzen/01asv/>.
- 2) Korkmaz, G., Ekici, E., Özgüner, F. and Ümit Özgüner: Urban multi-hop broadcast protocol for inter-vehicle communication systems, *Proc. of ACM VANET*, pp. 76–85 (2004).
- 3) Lochert, C., Scheuermann, B. and Mauve, M.:

Probabilistic aggregation for data dissemination in VANETs, *Proc. of ACM VANET*, pp. 1–8 (2007).

- 4) Yu, B., Gong, J. and Xu, C.-Z.: Catch-up: a data aggregation scheme for vanets, *Proc. of ACM VANET*, pp. 49–57 (2008).
- 5) Zhao, J. and Cao, G.: VADD: Vehicle-Assisted Data Delivery in vehicular ad hoc networks, *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 3, pp. 1910–1922 (2008).
- 6) Gupta, I., Renesse, R. V. and Birman, K.: Scalable fault-tolerant aggregation in large process groups, *Proc. of IEEE Int. Conf. on Dependable Systems and Networks (DSN)*, pp. 433–442 (2001).
- 7) Madden, S., Franklin, M. J., Hellerstein, J. M. and Hong, W.: TAG: a Tiny AGgregation service for ad-hoc sensor networks, *SIGOPS Oper. Syst. Rev.*, Vol. 36, No. SI, pp. 131–146 (2002).
- 8) Boulis, A., Ganeriwal, S. and Srivastava, M. B.: Aggregation in sensor networks: an energy accuracy trade-off, *Ad Hoc Networks*, Vol. 1, No. 2-3, pp. 317–331 (2003).
- 9) Papadopoulou, M. and Schulzrinne, H.: Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices, *Proc. of ACM MobiHoc*, pp. 117–127 (2001).
- 10) : MobiREAL シミュレータ. <http://www.mobireal.net/>.
- 11) Xu, Y., Lee, W.-C. and Mitchell, G.: CME: a Contour Mapping Engine in wireless sensor networks, *Proc. of IEEE ICDCS*, pp. 133–140 (2008).
- 12) Uchiyama, A., Fujii, S., Maeda, K., Umedu, T., Yamaguchi, H. and Higashino, T.: Ad-hoc localization in urban district, *Proc. of IEEE INFOCOM*, pp. 2306–2310 (2007).