

卒業論文 2017年度(平成29年)

自動車の位置情報ログデータを用いた  
駐車区画推定手法の提案

慶應義塾大学 総合政策学部  
豊田 安信

## 自動車の位置情報ログデータを用いた 駐車区画推定手法の提案

マイカー利用者による駐車場待ちや空き駐車場の探索により生じる交通渋滞は、日本の都市部や主要な観光地等において深刻な社会問題化している。

また、駐車場全体を目視で見渡すことの出来ない大型駐車場や階層型駐車場においても、施設への入り口や特定の店舗の近傍などの人気の高い区画に利用者が集中するため、空き状況の確認を試みる車両による渋滞が頻発する。一部の施設では超音波センサー等による常設筐体を用いて満空状況を管理する中央管理型システムが導入されているが、運営者が用意できるコストに依存している。

一方、近年では自律分散型制御を行うアーキテクチャも考案されている。これは利用車両に搭載された車々間通信端末によって位置情報を共有することで空き状況をセンシングするため、常設機器を必ずしも必要としない。しかし駐車場全体のネットワークモデルの事前の構築が必要になるため、結果として駐車場の運営主体による維持・運営を必要とするという欠点がある。

そこで、本研究では駐車場内の車両の位置情報のログデータから、特定のエリアにおける駐車場全体のネットワークモデルを構築するための手法を提案する。本手法を用いることによって、駐車場内での車両の振る舞いを予測し、駐車区画の位置関係の推定が可能になる。加えて、それらのデータから学習満車時の各駐車区画の収容台数を予測することが可能となる。

この手法を評価するため、駐車区画の人気度に差があり空き区画探索の徘徊が発生しやすい慶應義塾大学湘南藤沢キャンパス内駐車場をモデルケースとし、検証実験を実施した。独自に実装した交通シミュレータ上でキャンパスを模した駐車場の仮想地図を作成し、駐車場内の各駐車区画の数や座標、駐車枠の保有台数に対して十分な推定精度を得るために必要な車両台数や時間に関する評価を通じて、本研究が提案するアルゴリズムの妥当性が証明された。

本研究が提案するネットワークモデル構築システムと既存研究で用いられた手法を活用することで、路上常設のセンサーを用いることなく、駐車場全体の空き状況の把握・可視化・最適な区画へのナビゲーションを行うことが可能になり、利用車両の無駄な徘徊の低減や駐車時間の減少を見込める。加えて都市部や観光地の様な同様の問題が発生しやすい環境下においても、駐車場運営主体の枠を超えた空き状況の統合的な可視化の実現と、渋滞の低減に対する取り組みが期待される。

キーワード

1. ITS, 2. 位置情報, 3. プローブ情報, 4. 携帯電話網

慶應義塾大学 総合政策学部  
豊田 安信

## Inferring parking block from vehicle position data

The traffic congestion caused by searching for an empty parking lot has become a serious social problem especially in urban areas of Japan and major tourist spots.

Also, the congestion frequently occurs due to vehicles attempting to check the availability of a large parking area where you can not see the entire parking lot by visual inspection. Some of the stores have introduced Central management type fullness management system but it is expensive so that everyone cannot afford it.

On the other hand, an architecture that performs autonomous distributed control has been devised in recent years. This architecture does not necessarily require permanent equipment because it senses empty situation by sharing location information by Vehicle-to-Vehicle communication. However, it has the disadvantage of requiring the maintenance and operation of the parking area management entity because it is necessary to construct a network model of the entire parking area in advance.

Therefore, in this research, we propose a method for constructing the network model of the entire parking area in a specific area from the log data of the position information about vehicles in the parking area. By using this method, it is possible to predict the behavior of the vehicle in the parking lot and estimate the positional relationship of the parking section. Additionally, it is possible to predict the number of accommodated parking lots at full learning occasions from these data.

The verification experiment was carried out with Keio University Shonan Fujisawa campus parking area as a model case where there is a difference in the degree of the parking block of popularity and becomes vacant, and the loitering of the division search is easy to occur in order to evaluate this approach. We created a virtual map of a parking area imitating the campus and evaluated the generated network model on a proprietary traffic simulator. That proved the validity of the proposed algorithm.

By utilizing the proposed methods in this research, it becomes possible to grasp and visualize the vacancy situation of the entire parking area, and that leads to an optimized traffic volume. Even under similar environments such as urban areas and sightseeing spots where similar problems are likely to occur, there is an expectation to realize integrated visualization of vacancy situations beyond the framework of the parking management entity and efforts to reduce congestion.

Keywords :

1. ITS, 2. GPS, 3. Probe Vehicle Systems, 4. Mobile network,

Keio University, Faculty of Policy Management

Yasunobu Toyota

# 目次

第1章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的	2
1.3	本論文の構成	3
第2章	関連する技術や先行研究	4
2.1	駐車場空き状況センシング手法	4
2.1.1	敷設センサーによるセンシング	4
2.2	車両のセンサーを用いる駐車場内の最適化手法	6
2.2.1	VANET 技術	6
2.2.2	VANET を用いた駐車場の効率化手法	8
2.2.3	位置情報システムと携帯電話線網を用いたセンシング手法	9
2.2.4	位置情報ログデータを用いた道路地図の推定手法	10
2.3	車載アプリケーションプラットフォーム	14
第3章	提案手法	15
3.1	本提案手法の位置付け	15
3.2	用語	15
3.3	目的と仮説	16
3.3.1	本提案手法の目的	16
3.3.2	本研究の仮説	17
3.4	本提案手法の仮定と設計	17
3.4.1	位置情報の収集と提供	17
3.4.2	駐車区画の位置推定	18
3.4.3	駐車区画の駐車枠数の推定	19
3.4.4	道路地図の推定	19
3.5	フロー図	19
第4章	シミュレータの実装	21
4.1	設計	21
4.1.1	地図について	21
4.1.2	ノードとエッジ	21
4.1.3	車両について	22
4.1.4	時間経過と評価実験プログラムについて	26

4.2	クラス図 . . . . .	27
4.3	ネットワークモデルの推定 . . . . .	27
4.3.1	駐車区画座標の推定手法 . . . . .	27
4.3.2	駐車区画の有する駐車枠数の推定手法 . . . . .	29
4.3.3	駐車区画・交差点間を結ぶ道路の推定手法 . . . . .	30
4.4	実装環境 . . . . .	30
<b>第5章</b>	<b>評価</b>	<b>32</b>
5.1	仮定 . . . . .	32
5.1.1	運用環境のモデルケース . . . . .	32
5.1.2	条件設定 . . . . .	34
5.2	評価実験 . . . . .	35
5.2.1	評価実験1：駐車区画の位置推定 . . . . .	35
5.2.2	評価実験2：駐車区画が保有する駐車枠数の推定 . . . . .	38
<b>第6章</b>	<b>結論</b>	<b>42</b>
6.1	本研究のまとめ . . . . .	42
6.2	将来的な展望 . . . . .	42

# 目次

1.1	駐車場の空き待ち行列の発生に伴う周辺道路の渋滞の増加 <sup>1</sup>	1
1.2	導入が進められている駐車場空き待ち予約システム <sup>2</sup>	2
2.1	フラップ式センサーの設置された駐車区画 <sup>3</sup>	4
2.2	羽田空港国際線ターミナル駐車場に設定されているモニタリングセンサーノード <sup>4</sup>	5
2.3	監視カメラによる満空情報の認識システム <sup>5</sup>	5
2.4	タイムズ24株式会社 タイムズ駐車場検索 <sup>6</sup>	6
2.5	VANETのネットワーク図 <sup>7</sup>	7
2.6	R. Luらにより提案されたSPARK手法の概念図	9
2.7	MathurらによるParknet手法の概略図 <sup>8</sup>	10
2.8	k-Means法を用いる道路地図推定のフロー <sup>9</sup>	11
2.9	Trace Merging法を用いる道路地図推定のフロー <sup>10</sup>	12
2.10	カーネル密度推定を用いる道路地図推定のフロー <sup>11</sup>	12
2.11	k-Means法・Trace Merging法・KDE法の評価事例 <sup>12</sup>	13
3.1	本研究で定義するネットワークモデルの例	16
3.2	本提案手法での各ホストのコンセプト	18
3.3	本提案手法のフローチャート	20
4.1	1シーケンス中の <i>Car</i> インスタンスの振る舞いを表す図	24
4.2	誤差関数 $g(x)$ が返す位置情報と真値との差分の分布を表すヒストグラム	26
4.3	本シミュレータの各クラスの関係性	28
5.1	googleMaps上にプロットしたSFC内駐車区画のネットワークモデル	33
5.2	ネットワークモデル化したSFC	33
5.3	評価実験1が成功したケースにおける推定された駐車区画座標と真値の比較	37
5.4	利用車両の台数と推定の成功率の関係性	37
5.5	仮想地図内の最大車両台数と各駐車区画の推定充足率	39
5.6	仮想地図内の最大車両台数と各駐車区画の推定充足率	40

# 表 目 次

4.1	地図を表す <i>Map</i> クラス . . . . .	21
4.2	駐車区画・交差点を表す <i>Place</i> クラス . . . . .	22
4.3	仮想車両を表す <i>Car</i> クラス . . . . .	23
4.4	本シミュレータを表す <i>Simulator</i> クラス . . . . .	26
4.5	シミュレータの実行環境 . . . . .	30
5.1	SFC の仮想地図内における各区画の属性 . . . . .	34

# 第1章 序論

## 1.1 本研究の背景

日本において、都市部や観光地での交通渋滞は日常的に見られる光景である。特に人気の高い施設の周辺道路では、空きのある駐車場を求める車両によって大きな渋滞を引き起こすケースがある。

○観光周辺では、満車駐車場の広がりによって渋滞が延伸

■ 出雲大社周辺への進入経路と渋滞の状況

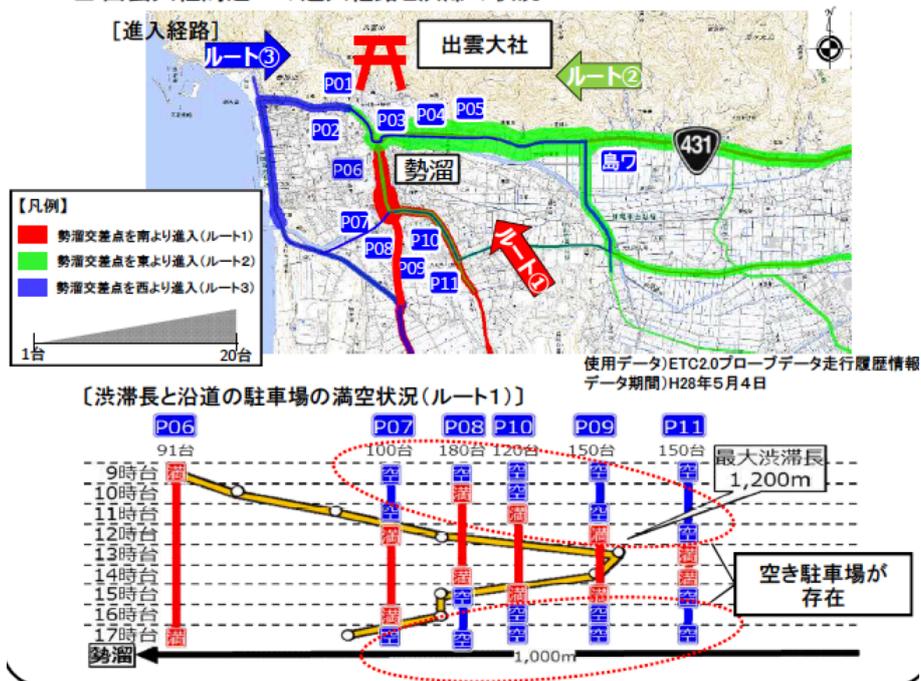


図 1.1: 駐車場の空き待ち行列の発生に伴う周辺道路の渋滞の増加<sup>1</sup>

国土交通省の調査 [1] によると、一部の人気観光地において駐車場待ちによる渋滞が多発している一方で、近傍には空きのある駐車場が存在するケースが報告されている。図 1.1 に駐車場の空き待ち行列の発生に伴う周辺道路の渋滞の発生をグラフに示す。

これらのような環境では、空きのある駐車場の可視化や利用者の誘導を行うシステムが求められていたが、近年では事前予約システムや空き待ち予約システムなどの試

<sup>1</sup>国土交通省の配布資料より引用 [1]

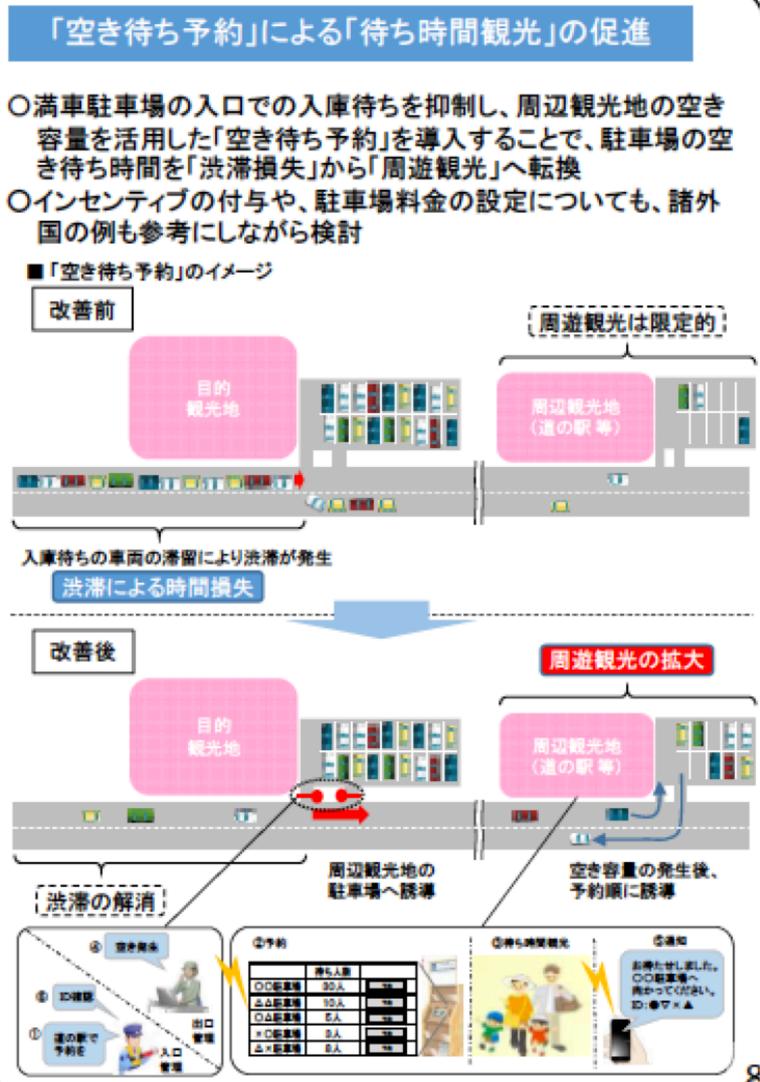


図 1.2: 導入が進められている駐車場空き待ち予約システム<sup>2</sup>

験的な導入が決まっている。その一例として、国土交通省は図 1.2 のように周辺駐車場への誘導から交通の分散と周遊観光の拡大を目指すシステムを提案している。

しかし、これらのプラットフォームを効果的に運用するためには、各駐車場のシステムの統合的な管理・運用が求められる。管理・運用コストの高さから常設機器の設置が困難なため二の足を踏む駐車施設も多い。駐車場は運営者が個人・民間・行政とまたがるため、それぞれの駐車場に統一した設備を設置するのは大きな障壁になる。

## 1.2 本研究の目的

既存の駐車場満空システムでは、運営主体による積極的な投資・運営・管理が必要であった。すなわち、駐車場の効率化の主体はあくまで運営者であったとも言える。

<sup>2</sup>国土交通省”観光地における渋滞対策”より引用 [1]

本研究では車両の位置情報と携帯電話回線網を用いて，駐車場全体のネットワークモデルを自律的に生成する手法を提案する．これにより昨今多方面で提案されていた車両のセンサーを用いる駐車場効率化手法を，運営主体の枠組みを超えて統合的に行うことが可能になる．

### 1.3 本論文の構成

本論文の構成を以下に示す．第2章では本研究に関連する技術や先行研究を述べる．第3章では本提案手法の目的と仮説，具体的なアプローチに関して述べる．第4章には評価実験を行うシミュレータの実装について記述する．第5章では評価実験の概要と結果，考察について述べる．第6章では本研究のまとめを記述する．

## 第2章 関連する技術や先行研究

ICT 環境の発達によって従来の敷設センサーに頼らない駐車場の効率化が多く研究されている。本章では駐車場の効率化手法を分類し、本提案手法が担う役割に関して記述する。

### 2.1 駐車場空き状況センシング手法

本節では様々な駐車場の空き状況センシング手法について記述する。

#### 2.1.1 敷設センサーによるセンシング

日本において従来より広く導入されている手法である。運営主体によって路側に常設された機器がセンシングを行う。そのため、管理・利活用・情報の開示にかかるコストを運営者が負担する場合が多い。これらのセンサーによって得られた満空情報についてどこまでの情報を開示し提供するかは運用主体の裁量で決まる。以下に主な敷設センサーを列挙する。

- フラップ式センサー  
都市部の地上型駐車場で一般に普及しているセンシング手法である (図 2.1)。利用車両の未払いを防ぐ働きも担っている。
- モニタリングセンサーノード  
大型の階層型駐車場で用いられている手法である (図 2.2)。それぞれの駐車枠上



図 2.1: フラップ式センサーの設置された駐車区画<sup>1</sup>

<sup>1</sup><https://www.photo-ac.com/main/detail/505568> より引用 最終確認日 2018 年 1 月 17 日



図 2.2: 羽田空港国際線ターミナル駐車場に設定されているモニタリングセンサーノード<sup>3</sup>

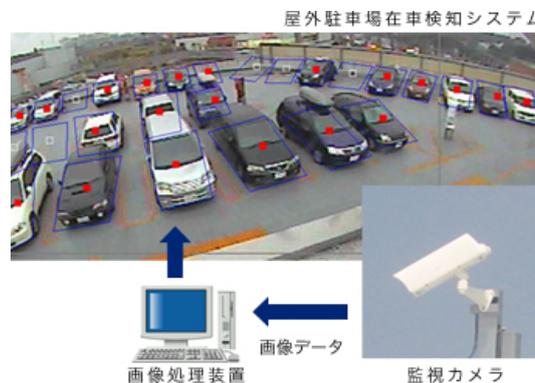


図 2.3: 監視カメラによる満空情報の認識システム<sup>4</sup>

部に常設機器を設置し空き状況をセンシングする．利用者は施設内に設置されたモニターで各駐車区画の空き状況を確認出来るケースが多い．

- 監視カメラを用いた画像認識

監視カメラの映像から各区画の満空情報を認識する手法である(図 2.3)．近年急速に導入されている．ステレオカメラを用いる手法も研究されており，山崎ら [2] によれば従来型のシステムよりも高い精度で車両を認識することが可能になるとされている．

運営主体によっては上記のセンサーで得られた満空情報を独自のプラットフォームでユーザーに公開する場合がある．例えばタイムズ 24 株式会社では，自社が運営している駐車場のたまなか満空情報を WEB サービスを用いてユーザーに公開している [3]．図 2.4 にタイムズ 24 株式会社による検索サービスのスクリーンショットを示す．一方でユーザーが同じ地域の空き駐車場を検索するためには，運営主体ごとにそれぞれ独立した WEB サイトにアクセスする必要があり利便性を欠く．また設置・運営に関わる

<sup>3</sup>羽田空港国際線ターミナル駐車場にて 2017 年 11 月撮影

<sup>4</sup>株式会社ニチゾウテック 駐車場誘導管制設 <http://www.nichizotech.co.jp/products/04parking/index.html> より引用 最終確認日 2017 年 1 月 17 日



図 2.4: タイムズ 24 株式会社 タイムズ駐車場検索<sup>5</sup>

コストを全て運営主体が担わなくてはならないため、運営コストを割くことが出来ない学校のような非営利団体が運営する駐車場での導入は困難である。

## 2.2 車両のセンサーを用いる駐車場内の最適化手法

従来、駐車場内の満空管理や車両の誘導は路側に設置されたセンサーやインジケータを用いる手法が一般的であった。しかしながら近年ではVANETや衛星測位システム、携帯電話回線網を用いる手法が提案されている。

### 2.2.1 VANET 技術

本節ではEliasらによるサーベイ [4] を引用し、多くの駐車場内の効率化システムを実現する技術であるVANETについての概略について述べる。

#### 概要

車々間・路車間で構成されるアドホックネットワークを、一般してVANET(Vehicle Adhock Network)と呼称する。VANETに参加する車両には車載通信装置(OBU)が装

<sup>5</sup><https://times-info.net/> より引用 最終確認日 2017 年 1 月 17 日

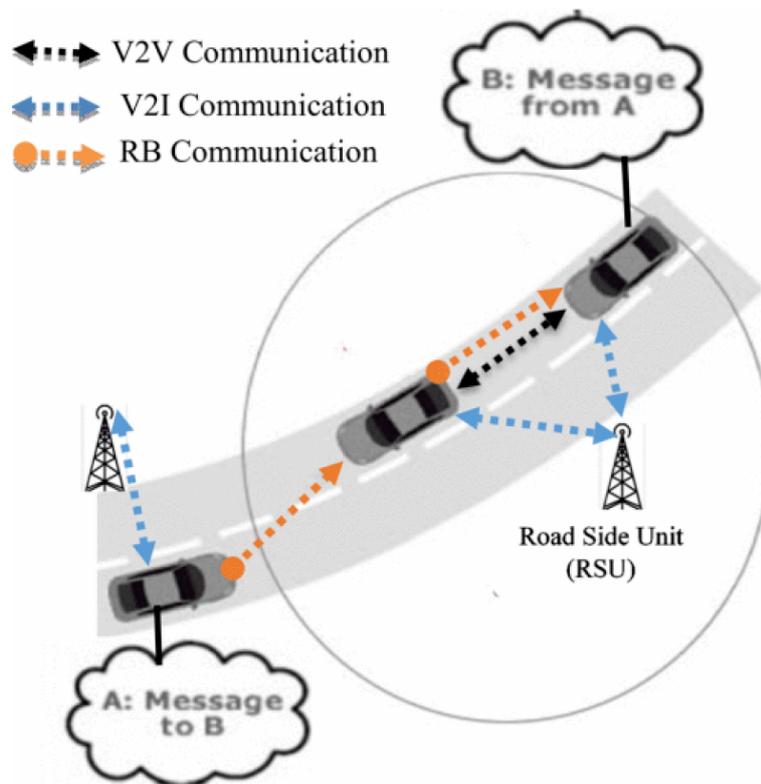


図 2.5: VANET のネットワーク図<sup>6</sup>

備され、他の OBU や路側装置 (RSU) との無線通信が可能である。図 2.5 に VANET での通信の概要を示す。

車両の OBU 同士の通信は V2V (Vehicle to Vehicle) 通信と呼ばれ、タイムクリティカルな危険予知情報の伝達に用いられる。

一方で、路側に設置された RSU と OBU との通信は V2I (Vehicle to infrastructure) 通信と呼ばれる。これは主に RTK-GPS[5] のような数センチ単位の正確な位置情報を車両側に提供するプロトコルや、交通信号機や指示板の情報の伝達に用いられる。

また、OBU-RSU を伝播して情報を伝える通信をルーティングベース (RB) 通信と呼ぶ。RB 通信を用いて駐車場内の満空状況を車両のみで伝達する手法が山下らによって提案されている [6]。

#### 想定する無線通信規格

VANET は各国の法規制や電波環境に併せて様々な無線通信規格での運用が議論されている。

- IEEE 802.11p[7]

IEEE によって定められた ITS 用の国際的な無線通信規格である。基本的には

<sup>6</sup>Elias らの資料 [4] より引用

5.9GHz 帯を用いる。米国を中心に標準化・普及が行われているが、日本においてはノンストップ自動料金支払いシステム (ETC) に代表される狭域通信 (DSRC)[8] 専用として利用される ARIB STD-T75[9] と、周波数帯が近いため導入が遅れている [10]。

- 携帯電話回線

ITU によって標準化されている 2G, 2.5G, 3G, 4G などの携帯電話回線網 [11] は信頼性の高いセキュリティと広いカバレッジを得られるメリットがある反面、リアルタイム性に優れないというデメリットがある。一方で、近年では次世代規格である低遅延・多数同時接続を目標とした 5G の策定も進んでおり、VANET を用いた安全運転支援や自動運転車両の遠隔制御などでの活用が期待されている [12]。

- Unified wireless access

ISO/TC/204/WG16[13] にて既存の無線通信を統合的に扱うための標準仕様として策定されたのが CALM M5[14] である。これは IEEE802.11p をベースに携帯電話回線の技術を加えて構成されており、スループットの増加や冗長化が可能になっている。

## 2.2.2 VANET を用いた駐車場の効率化手法

VANET には安全運転支援システムやプラトウニングなどへの応用例が目立つが、駐車場の効率化手法としても取り入れられつつある。本節では VANET を基盤技術として用いる関連研究に関して述べる。

### SPARK

R. Lu らによって提案された SPARK[15] は、VANET を介して全ての利用車両に対して駐車場所を指定していくプラットフォームである。図 2.6 は本手法が用いる V2I 通信を簡潔に表したものである。中央システムが利用車両に電子チケットを発行して駐車区画の占有権を与える役割の他に、その区画へのナビゲーション機能や車両盗難防止機能なども提案している。SPARK では、電子チケットを発行した車両がそこに駐車したと仮定することで、中央システムが駐車場全体の満空情報の監視・提供することを想定している。しかしながら、路側センサーの設置と中央システムの運営を運営主体が行う必要があるという従来型満空センシングアーキテクチャが抱える問題をそのまま継承している。

### V2V 通信のみで情報を伝達する手法

山下らは、車両の近接センサーによって得られた近傍駐車枠の満空情報を VANET を用いて伝達し、ネットワークに参加している車両内で駐車場全体の満空情報を共有

---

<sup>8</sup>SPARK の資料 [15] より引用

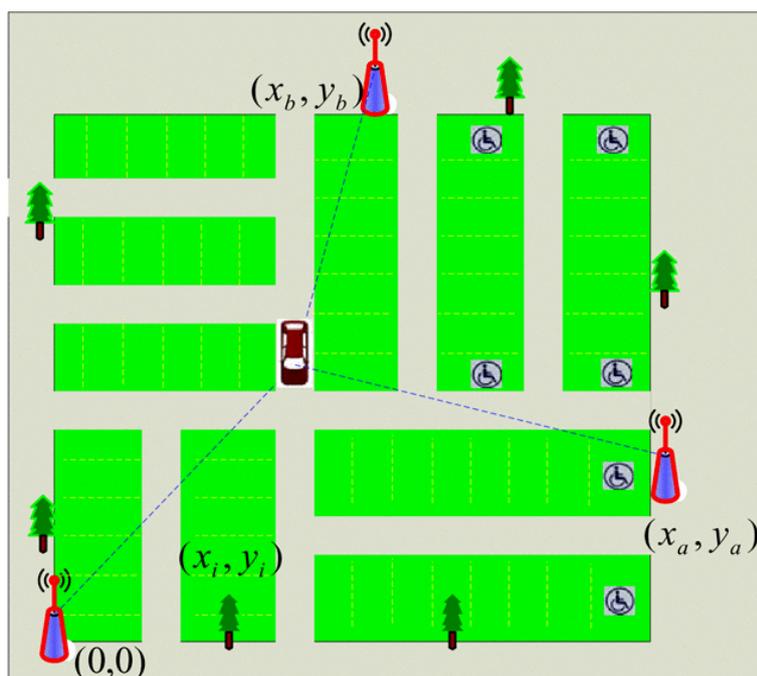


図 2.6: SPARK 手法では RSU を用いた V2I 通信を活用することで, 利用車両への駐車枠の誘導や情報提供を可能にしている<sup>8</sup>

する手法を提案している [6]。各車両が路面に書かれた駐車区画番号を車載カメラによる画像認識することで位置情報を特定することを想定している。中央のシステムに頼ること無く駐車場内の効率化が図れる一方、駐車場内のネットワークモデルを前提知識として共有しているため、運営者による詳細なネットワークモデルの提供が不可欠であると言える。

### VANET による位置測位と路側から情報共有によるナビゲーションシステム

孫らは、路車間通信によって車両の位置情報から割り出された最適な駐車区画への経路を利用者にナビゲーションする手法を提案している [16]。中央のシステムが RSU を介して車両の位置情報を収集し、満空情報と駐車成功度を車両に広報する。これらの情報を基に、新しく駐車場内に進入した車両に対して、最適な駐車区画の割当・ナビゲーションが行われる。システムを搭載した車両が少ない場合でも駐車までの時間が短縮される点が特徴である。一方、このアーキテクチャでも、事前に中央システム・利用車両が駐車場内のネットワークモデルを把握している必要がある。

### 2.2.3 位置情報システムと携帯電話線網を用いたセンシング手法

近年、位置情報システムを用いたサービスやアプリケーションがより身近なものになっている。位置情報システムで最も広く利用されているのが GPS に代表される全球測位衛星システム (GNSS) である。GPS の測位位置の誤差は通常 4.9M ほどであり

[17]，近年では準天頂衛星システム (QZSS) による補正を受けることで理論上 1M 以内の高精度を見込める [18]。また，広帯域な携帯電話回線を安価に導入できる環境が整ったことで，携帯電話回線を備えた車載機器を用いて位置情報の収集を行う手法も一般的になっている。

本項では位置情報システムと携帯電話回線網によって満空情報を推定する手法について述べる。

### 車載センサーと携帯電話回線網を用いた路上駐車枠の空き情報検知手法

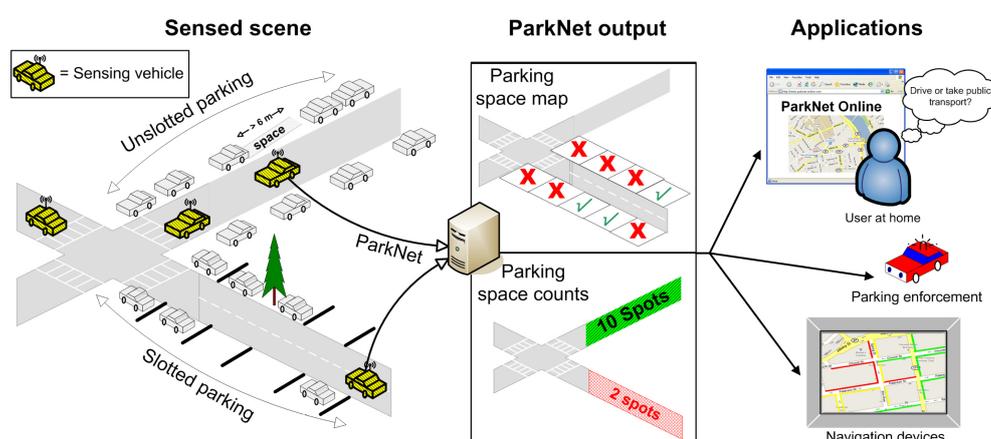


図 2.7: Mathur らによる Parknet 手法の概略図<sup>9</sup>

Mathur らによる Parknet プロジェクトでは，車載センサーと GPS システムを用いて路上駐車枠をセンシングする手法を提案している [19]。車両から得られたそれぞれの駐車枠の満空情報を，携帯電話回線網によって中央サーバーに収集する。それらのリアルタイムな情報から，中央サーバーが都市全体の路上駐車場のネットワークモデルを構築する。

また，Coric らは Parknet 手法に駐車禁止枠に対する違法駐車検知機構を取り入れたアーキテクチャを提案している [20]。この駐車禁止検知システムでは，90 パーセント程度の推定精度が確認されている。

これらの手法の優位点は，路上に設置したセンサーに頼ること無くネットワークモデルの構築を動的に行える点にある。しかし，路上駐車場のみにフォーカスされており，敷地内駐車場に関してはまだ取り組むべき余地を残している。

#### 2.2.4 位置情報ログデータを用いた道路地図の推定手法

スマートフォンやカーナビゲーションの移動ログデータの活用方法の一つとして，移動データから地図そのものを生成・更新する取り組みが近年注目されている。ナビゲー

<sup>9</sup>Parknet の資料 [19] より引用

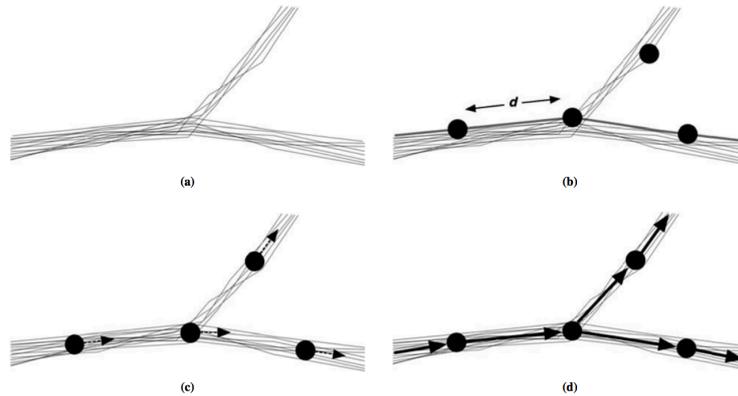


FIGURE 1 Map inference with  $k$ -means algorithm: (a) raw traces, (b) initial seeds, (c) updated means with bearing, and (d) final graph after means are linked.

図 2.8:  $k$ -Means 法を用いる道路地図推定のフロー <sup>10</sup>

ションサービスの提供者にとって、地図を作成する費用を軽減出来る点だけではなく、急な通行止めや道路情報の変更リアルタイムに柔軟な対応を可能にする点も大きなメリットになる。

### 道路地図推定手法の分類と比較

位置情報ログデータを用いる道路地図の推定に関して、様々な手法での提案が行われているが、Biagioni らによるサーベイ [21] では大別して三つのアプローチに分類している。本節ではそれらの手法の分類と特徴について述べる。

- $k$ -Means 法

$k$ -Means 法は Forgey らによって 1965 年に提唱されたクラスタリング手法である [22]。近年では多くの応用分野に取り入れられている。道路地図の推定という文脈では、Edelkamp らによって提案された手法 [23] が著名である。

図 2.8 に具体的なフローを示す。まず始めに、加工をしていない移動ログデータに初期シードを一定の間隔で設置し、 $k$ -Means 法を用いてクラスタリングする。その後生成されたクラスターシードを、移動実績のある他のシードと結ぶことでエッジを生成する。これを繰り返すことで生成されたエッジから道路地図を推定することが出来る。

- Trace Merging 法

Cao らによって提案された手法が Trace Merging 法である [24]。この手法のフローを図 2.9 に示す。始めに、収集した位置情報ログデータをそれぞれ有向グラフとして表現する。次にこのグラフから一つを任意に選択し、このグラフのエッジの近傍で方向が一致するノード数をエッジの重みとして加算していく。最終的

<sup>10</sup>Biagioni らの資料 [21] より引用

<sup>11</sup>Biagioni らの資料 [21] より引用

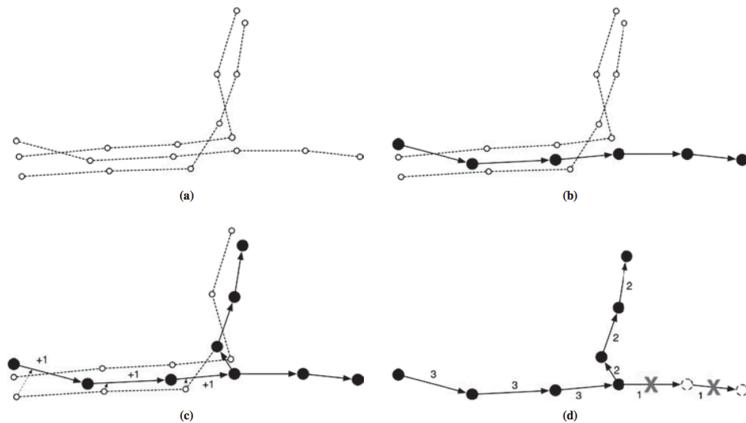


FIGURE 2 Trace-merging approach: (a) raw GPS traces, (b) new edges added, (c) weights of matching edges incremented, and (d) edges with insufficient weight removed.

図 2.9: Trace Merging 法を用いる道路地図推定のフロー <sup>11</sup>

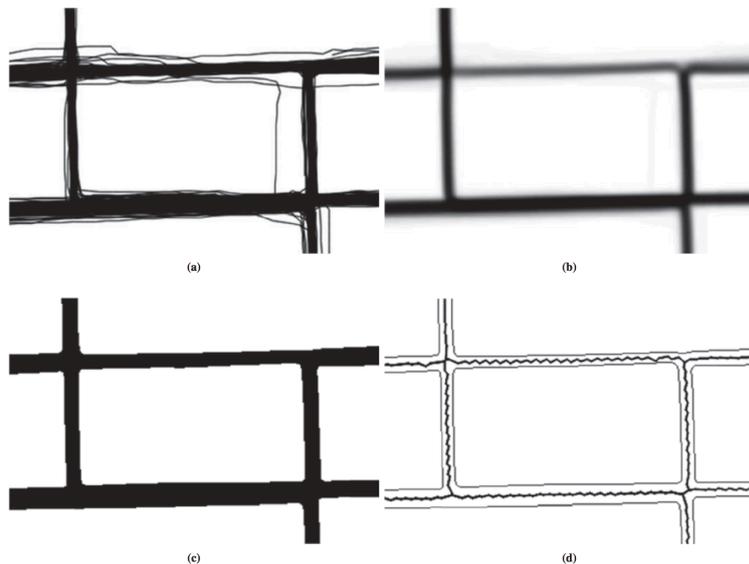


FIGURE 3 Map inference based on KDE (4): (a) raw traces, (b) KDE computed, (c) threshold applied and road or nonroad cells identified, and (d) road outlines and centerlines computed.

図 2.10: カーネル密度推定を用いる道路地図推定のフロー <sup>12</sup>

に残ったエッジから重みの少ない物を削除する．これにより，道路地図全体のネットワーク図が得られる．

- カーネル密度推定法

Davies らによってカーネル密度推定 (以後 KDE 法) [25] を用いて道路地図推定を行う手法が提案されている [26] ．

KDE 法を用いるフローを図 2.10 に示す．まず，元の移動ログ群から KDE 法を用いて 2D ヒストグラムを生成し，一定の閾値で二階長化する．生成された画像

<sup>12</sup>Biagioni らの資料 [21] より引用

から道路の中心値群を結んで道路全体の地図を推定する。



FIGURE 5 Raw data and results from three implemented algorithms in areas with high and low GPS error: (a) raw data, high-error sample; (b) raw data, low-error sample; (c) Davies algorithm, high-error sample; (d) Davies algorithm, low-error sample; (e) Cao algorithm, high-error sample; (f) Cao algorithm, low-error sample; (g) Edelkamp algorithm, high-error sample; and (h) Edelkamp algorithm, low-error sample.

### 図 2.11: k-Means 法・Trace Merging 法・KDE 法の評価事例<sup>13</sup>

こうした三つの手法を，実環境データにて評価した事例図 2.11 に示す。

ノイズの少ないログデータを用いた場合，三手法共に有効な結果を生成しているが，KDE 法では移動量の少ない動線が捨象されやすいことが見て取れる。

一方，ノイズが多いログデータを用いた場合には，Trace Merging 法と k-Means 法では有意な類推が出来てない。他方，KDE 法は捨象されている動線があるものの，真値に近い道路地図を推定していると判断可能である。

また，Biagioni らは各種法の実行時間に関しても言及している。100 パターンの移動ログをデータセットとした場合，Trace Merging 法が 2.5 時間の処理時間を要した一

<sup>13</sup>Biagioni らの資料 [21] より転載

方で、k-Means 法では 73 秒、KDE 法では 8 秒と手法間で大きく差が開いている。加えて 899 パターンのデータセットを与えた場合において各手法の実行時間は、Trace Merging 法では 2.5 日、k-Means 法では 15 分、KDE 法では 25 秒になった。締めくくりに Biagioni らは KDE 法をベースとして各手法を組み合わせた新たな手法の提案について示唆している。

### 三つの手法をあわせたアプローチについて

Biagini らは上記の三つの手法を組み合わせた新しいアプローチについての発表を行っている [27]。これによると KDE 法をベースとしたハイブリッドアプローチを用いることによって、より真値に近い道路地図の推定が可能になるとされている。

## 2.3 車載アプリケーションプラットフォーム

従来、自動車の情報端末はカーナビゲーションシステムとカーオーディオがその役割を担っていた。近年ではニーズの多様化や ICT 環境の発展によって、多くの情報からユーザーをサポートする役割も併せて求められるようになり、より高次元化された IVI(車両情報端末) と総称される端末が登場している [28]。多くの IVI プラットフォームでは、位置情報・インターネットへのアクセス機構を備えており、自動車の情報を扱うための標準仕様の策定が活発である [29]。例えば、WEB 標準化団体である W3C からは Vehicle API<sup>14</sup> が提案されているほか、自動車メーカー各社からはスマートフォンとの連携機能を標準化する SmartDevicelink<sup>15</sup> が提案されている。

---

<sup>14</sup><https://www.w3.org/TR/vehicle-information-api/>

<sup>15</sup><https://smartdevicelink.com/>

## 第3章 提案手法

### 3.1 本提案手法の位置付け

本章では、2.2.2 項にて述べた駐車区画の満空情報を共有する手法群や利用車両のナビゲーションを行う手法群を運用する際に、前提条件となる駐車区画の位置関係や収容台数を表すネットワークモデルの自律的な推定手法について述べる。

本研究では、車両の位置情報や駐車情報の収集に際しては2.3 節で述べたような車載アプリケーションプラットフォームによって行うと想定する。これにより VANET のような専用機器を用いる手法と比較して普及の容易度を高めることが出来る。

2.2.4 項で述べたように、KDE 法を中心とした様々なアルゴリズムを用いることで、位置情報ログデータから道路地図は安定して生成出来ることが明らかにされている。しかしながら、駐車場のネットワークモデルを構築するためには駐車区画の位置情報、駐車枠数のパラメーターが必要になる。本研究ではこれらのパラメーターの推定手法に関して検討する。

### 3.2 用語

本論文内で用いる用語について、以下に定義する。また、図 3.1 に本論文で定義する用語とネットワークモデルの概念図を示す。

- 駐車区画：車両が駐車可能な枠が隣接する場所を指す。一つの駐車区画は駐車枠を単数あるいは複数持つ。
- 駐車場：駐車区画群を持つ施設全体を示す。
- 駐車枠：車両一台を駐車出来る箇所を指す。本論文、提案手法では車両の大きさによる駐車枠の違いは考慮しない。
- ネットワークモデル：座標・駐車枠の数をパラメーターに含む駐車場内の駐車区画を表すノードと、それらを結ぶ道路を表す重み付きエッジを含むデータ構造と定義する。ネットワークグラフの形で視覚的に表現することが出来る。
- 交差点：複数の道路が接続されている地点を差す。駐車場をネットワークモデルとして表した場合、駐車枠を持たないノードが交差点にあたる。

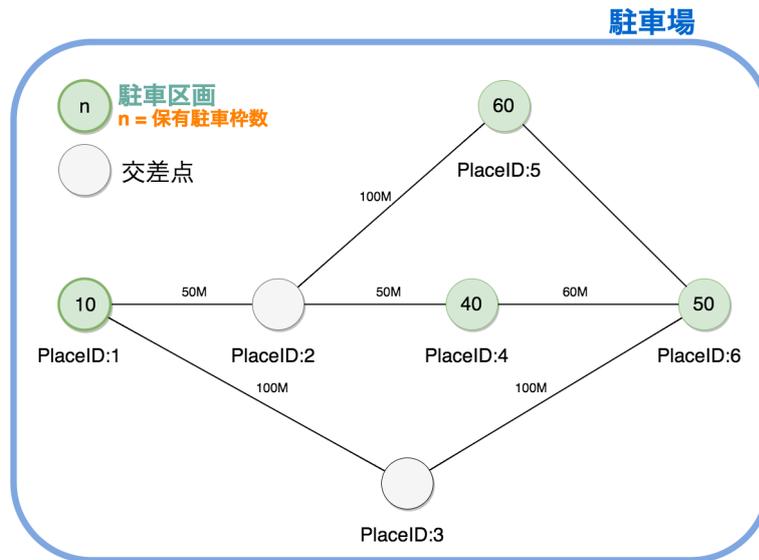


図 3.1: 本研究で定義するネットワークモデルの例

### 3.3 目的と仮説

#### 3.3.1 本提案手法の目的

本提案手法では既存の駐車場内効率化手法・ネットワークモデル構築手法の諸問題に以下のように取り組む。

##### 運営主体に依存しないアーキテクチャ

3.1 節でも述べたように、駐車場内に常設されたセンサーにより満空情報を管理する従来のアーキテクチャでは、運営主体が負担できるコストに実施性を依存していた。そのため余分な運用コストを費やすことの出来ない駐車場施設では、利用者の利便性の向上が困難であった。

また、2.2.2 項で指摘したように、車両側のセンサーを活用する手法では、ネットワークモデルの事前構築が必要という点で、結果的に運営主体による維持・管理が必要であった。

本提案手法では、運営主体に依らない駐車場効率化手法を実現可能にするためのネットワークモデルの自律的な推定を目指す。

##### 駐車場を含めたネットワークモデルの構築

2.2.4 節で述べた道路地図推定手法では、駐車場の推定には焦点が当てられていなかった。本研究では駐車場の位置推定を可能にするアーキテクチャを提案する。

## 敷設センサーやRSUが不要

第2章でも述べたように、既存の多くの駐車場最適化手法では路上側にセンサーや通信機器(RSU)を設置する必要があった。利用車両の位置情報測位システムと携帯電話回線網を用いる本提案手法では、路側に常設する機器が必ずしも必要にならない。そのため、駐車場の運営者による投資が難しい非営利施設の駐車場でも、駐車場効率化サービスを運用することが可能になる。

### 3.3.2 本研究の仮説

本研究では、3.1節で述べたような既存の駐車場効率化手法の諸問題の解決に取り組む。自動車の位置情報を収集・解析することで、駐車区画の設定や最大収容台数を駐車場ごとに設定したり路側にセンサーを常設することなく、駐車区画の空き状況を収集・可視化することが出来るようになることを予想している。

## 3.4 本提案手法の仮定と設計

### 3.4.1 位置情報の収集と提供

本提案手法には駐車場を利用する車両の位置情報を収集する機構が必要になる。ここでは位置情報収集システムに関して述べる。

#### 車載器とサービスクライアント

2.3節で述べたような自動車用アプリケーションプラットフォーム上で動作するサービスクライアントを利用者の車両で動作させる。サービスクライアントは携帯電話回線網を用いて、自動車の位置情報をサービスサーバーに定期的に送信する。また、車両が駐車した場合にエンジンの停止を検知して駐車シグナルを同時に送信するものとする。

#### サービスサーバー

本提案手法では、情報を収集しネットワークモデルを推定するサービスサーバーの実装を想定する。サービスクライアントとサービスサーバ間はインターネットを介して接続される。

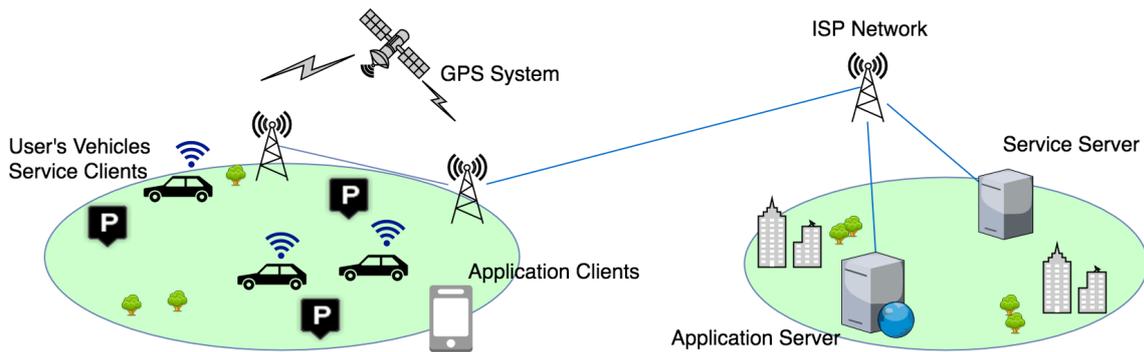


図 3.2: 本提案手法での各ホストのコンセプト

### ネットワークモデルを利用するアプリケーション

満空情報の可視化や駐車場のナビゲーションを行う様々なアプリケーションを運営するサーバーをアプリケーションサーバーと定義する．サードパーティによるアプリケーションのリリースを想定し，アプリケーションサーバーは，サービスサーバーとは別に構築する．アプリケーションサーバーはサービスサーバーからネットワークモデルや車両の位置情報の提供を受ける．

アプリケーションのクライアントは必ずしも車載された端末である必要はなく，WEBサービスやスマートフォン上で動作するアプリケーションを介して利用者にサービスを提供する形を想定する．

#### 3.4.2 駐車区画の位置推定

一定期間において収集された利用車両の駐車シグナルを受けた座標群から，X-Means法を用いてクラスタリングを行い，各クラスターの重心値を駐車区画の位置座標と推定する．

#### X-Means 法

X-Means 法とは Pelleg らによって提案された非階層型クラスタリングアルゴリズムである [30]．クラスタ数の決定を行うために発見的方法に頼る必要があった k-Means 法の発展形として考案された手法で，クラスター数をデータセットに合わせて推定し，データ群を適切な数のクラスターに分けることが出来る．

X-Means 法の基本的なアルゴリズムを下記に簡便に表す．

1.  $n$  個の  $p$  次元行列のデータセットを用意する．
2. 始めに  $k = 2$  として k-Means 法を実行し 2 つのクラスターにデータを分割する．

3. 各クラスタをベイズ情報量基準 (BIC) などの情報量基準尺度を用いて評価し, 2 クラスタに分けることが適当と判断できる場合, そのクラスタ内で 2. を実行する. そうでない場合にそのクラスタの分割を終了する.
4. 全てのクラスタの分割が終了するまで 2. と 3. を継続する.

本提案手法では, Pelleg 手法を一部改良した石岡手法 [31] を採用した. 石岡手法ではクラスタリングしたデータ群が必ずしもガウス分布していないことを考慮しているため, 多様なパターンのデータへの汎用性が高い.

### 3.4.3 駐車区画の駐車枠数の推定

3.4.2 項で推定された駐車区画の座標を基に, 駐車ログ群がどの駐車区画で行われたものを推定し, 同一単位時間中  $t$  の最大駐車台数をその駐車区画の保有駐車枠数とする. 駐車区画の枠数  $S_p$  は以下のような数式で表される.

$$S_p = \max\{f(t), t = 1, 2 \dots, \max\{t\}\} \quad (3.1)$$

$$S_p \leq G_p \quad (3.2)$$

$$G_p = p \text{ の保有する駐車枠数の真値} \quad (3.3)$$

$$f(t) = \text{単位時間 } t \text{ 時の } p \text{ に駐車している車両台数} \quad (3.4)$$

$$t = \text{単位時間} \quad (3.5)$$

$$\max\{t\} = \text{推定対象とするデータセットの時間} \quad (3.6)$$

### 3.4.4 道路地図の推定

道路地図の生成には, 走行ログデータ (駐車時以外の位置情報ログデータ群) を用いる. 走行ログデータから, 2.2.4 節にて述べた Davies らによる手法 [26] などを用いて道路地図を推定する.

生成された道路地図上に 3.4.2 節で求められた駐車区画を割り当てる.

## 3.5 フロー図

図 3.3 に本提案手法のフローを表す. 以下に具体的な動作を列挙する.

### 1. データの収集

利用車両の車載端末から位置情報を収集する. 推定対象とするデータセットの時間  $n$  の間に, 駐車場  $P$  に進入した車両から収集したネットワークモデル推定に用いるデータセットを  $D_p$  とする.

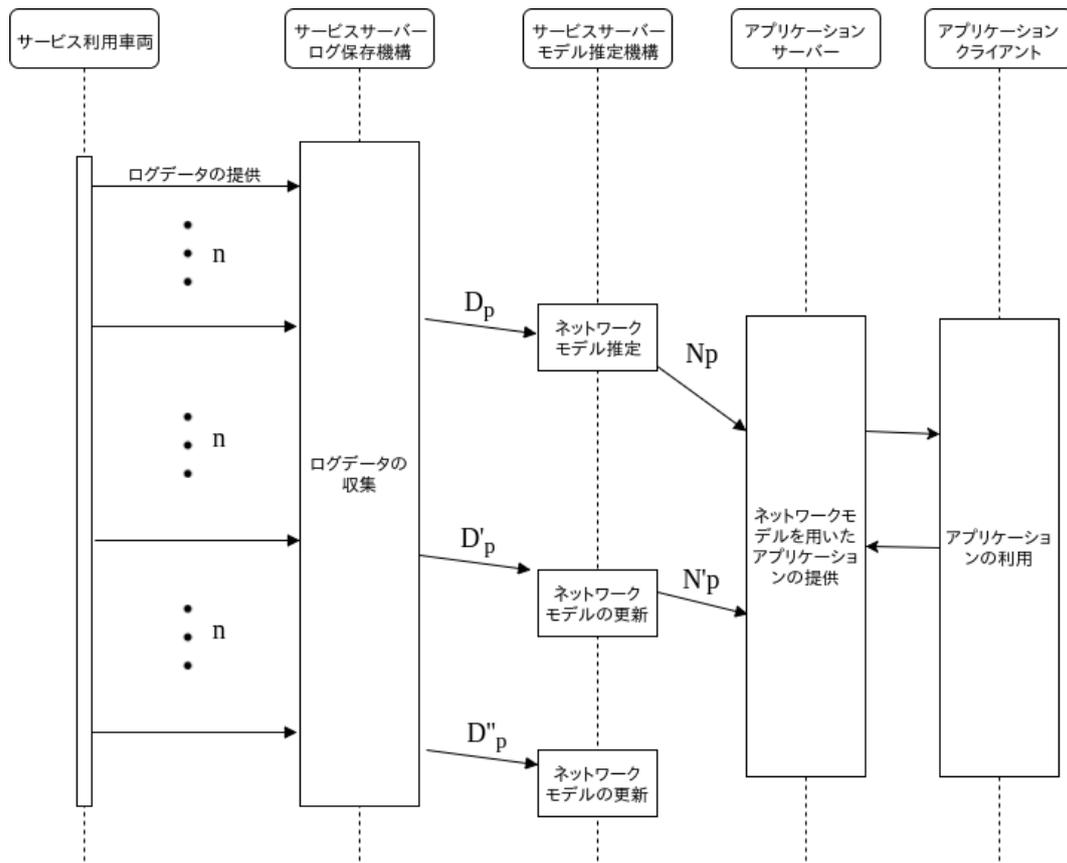


図 3.3: 本提案手法のフローチャート

## 2. ネットワークモデルの推定

データセット  $D_p$  に対して第 3.4.2 項にて述べた手法で解析を行う．その結果を推定されたネットワークモデル  $N_p$  とする．

## 3. ネットワークモデルの提供

ネットワークモデルを利用するアプリケーションに  $N_p$  を提供する．以降， $n$  が経過するたびに 1. と 2. を繰り返す，あらたに得られたデータセット  $D'_p$  から推定した結果  $N'_p$  を駐車場  $P$  のネットワークモデルとする．

## 4. アプリケーションの提供

サービスサーバーに蓄積された各駐車場のネットワークモデルを，様々なアプリケーションが使用することを想定する．図 3.3 では駐車場  $P$  のネットワークモデルをアプリケーションサーバーが利用する．アプリケーションサーバーはアプリケーションクライアントと相互に通信し，満空状況の可視化のような駐車場の効率化サービスを提供する．

## 第4章 シミュレータの実装

本提案手法の妥当性を検討する評価実験を行うために、シミュレータを実装した。本章では本シミュレータの構成と、モデルケースとする慶應義塾大学湘南藤沢キャンパスの駐車場環境について概説にする。

### 4.1 設計

本シミュレータは、事前に作成した仮想地図上を仮想車両群が走行し、各車両の位置情報の履歴のみを用いてネットワークモデルの推定を行うものである。

#### 4.1.1 地図について

本シミュレータでは仮想地図を *Map* クラスとして実装する。表 4.1 に主要なプロパティとメソッドを列挙する。

class Map	
プロパティ名	概要
places	保有する Place インスタンスの配列
parkings	保有する Places インスタンスの中で、駐車区画であるものの配列
graph	networkx の Graph インスタンス
paths	すべてのノードからノードへの最短経路を収納した二次元配列
メソッド名	概要
gen.Json	事前に作成した地図の構成を記した JSON 形式のテキストファイルから Map インスタンスを初期化する
choiceDest	places に記録された人気度から Car インスタンスの目的地となる Place インスタンスを返す

表 4.1: 地図を表す *Map* クラス

#### 4.1.2 ノードとエッジ

*Map* インスタンスは、駐車区画や交差点を示す“ノード”，それらを結ぶ道路を示す“エッジ”によって構成された重み付きグラフとして表現される。これは本研究の提案手法で推定するネットワークモデルと同一のものを示す。各要素の位置関係や接続関係は NetworkX<sup>1</sup> によって実装する。

<sup>1</sup>ネットワークグラフを扱うライブラリ。https://networkx.github.io/

## ノード

ネットワークモデルにおいて駐車区画及び交差点を表す“ノード”は *Place* クラスとして実装される．表 4.2 に *Place* クラスの主要なプロパティとメソッドを列挙する．

*Place* クラスは緯度・経度によって表現される位置情報 (*XY* プロパティ) を有する．本シミュレータではその情報に伴って仮想車両の位置情報が決定される．このフローに関しては第 4.1.3 項にて詳細に記述する．

また，各 *Place* インスタンスには保有する駐車枠数の情報が付与される．これは仮想車両が駐車・移動を行う度に増減する．

加えて，各インスタンスには人気度 (*popularity* プロパティ) を付与する．この人気度に則って車両の目的地が決定される．

class Place	
プロパティ名	概要
id_	Map インスタンスが持つ Place インスタンスを一意に決定する
capacity	Place インスタンス上に駐車可能な車両の数 交差点のインスタンスには 0 が挿入される
availability	現在の空き駐車枠の数 交差点のインスタンスには 0 が挿入される
isfull	満車かどうかを表す 交差点のインスタンスでは False が常に入る
XY	Place インスタンスの位置情報を (経度, 緯度) の配列で示す
popularity	駐車区画の人気度を示す値 0 ~ 100 の整数が入る
passage_cars	通過している Car インスタンスが挿入される配列
parking_cars	駐車している Car インスタンスが挿入される配列
メソッド名	概要
add_passage	車両が通過している際に実行される
add_park	車両が駐車しようとしている際に実行される 駐車可否を bool 型で返す
remove_car	車両がこのインスタンス上から立ち去る際に実行される

表 4.2: 駐車区画・交差点を表す *Place* クラス

## エッジ

道路を表すエッジは *Place* インスタンス間を 1:1 の関係で結ぶ．エッジには距離を表す指数 “weight” が付与されている．本シミュレータでは，先に述べた NetworkX の Edge インスタンスとして表現する．

### 4.1.3 車両について

本実装では仮想車両を *Car* クラスとして表現する．本シミュレータでは複数の *Car* インスタンスを作成し，移動ログデータを記録・解析する．図 4.3 に *Car* クラスの主要なプロパティとメソッドを記述する．

class Car	
プロパティ名	概要
id_	Simulator インスタンスが持つ Car インスタンスを一意に決定する
firstplace	Car インスタンスが生成された時に現在地とする Place インスタンス
destination	目的地となる Place インスタンスの id_
confusion	destination ではない駐車区画を通過する際に駐車を強行する確率 0~1 の float 型
path	destination までの最短経路 Map インスタンスが持つ path から取得する
isparked	駐車しているかどうかを表す bool 型
parking_counter	駐車するシーケンス数 Simulator クラスで定められた範囲内で決定される int 型
position	現在居る Place インスタンス
XY	現在居る位置情報が (経度, 緯度) の配列で挿入される
isonedge	現在道路上 (エッジ) にいるかどうか bool 型
error	位置情報の誤差の大きさを決定する値 float 型
メソッド名	概要
setPath	目的地までの経路を生成し path に代入する
nextseq	この車両のシーケンスを進める 下記の各メソッドを実行する
move_to	次に移動する Place インスタンスを引数に取り, Car インスタンスのプロパティの編集する
arrive_at	新しい Place インスタンスに到着した時に実行される
try_park_at	駐車区画の Place インスタンスに到着した際に駐車を試みる 成功の可否を bool 型で返す
park_at	try_park_at で駐車が成功した際に実行される
calcPosition	毎シーケンスごとに位置情報を Simulator インスタンスに渡すための関数 error の値に基づいて誤差を生成する

表 4.3: 仮想車両を表す *Car* クラス

### *Car* インスタンスの振る舞い

本項ではシミュレータ上における *Car* インスタンスの振る舞いについて記述する。*Car* インスタンスは 1 シーケンスが進むごとに仮想地図内を移動していく。その過程を図 4.1 に示す。

車両は 1 シーケンスごとに定められた台数が場内に入場する。この時の開始地点は *Map* インスタンスが持つ *Place* の内  $id_ = 0$  の地点とする。各車両は識別コード ( $id_$ ) の若い車両から順番に、1 シーケンスごとに進入・移動・駐車・退出を行う。

- 移動

場内に入場した車両は、それぞれに固有に与えられた目的地を目指して、定められた経路に則って移動する。なお、本シミュレータではダイクストラ法 [32] を用いて最短経路を計算する。エッジ上を移動している車両は 1 シーケンスごとに  $1[m]$  を移動する。車両の移動速度  $v$  は  $v = 1[m/s]$  である。

- 駐車

目的地となる駐車区画に到着した車両は必ず駐車を試みる。既に駐車している車両で駐車枠が占領されている場合、再度目的地の設定を行う。この時、同様に経路も再設定される。以後駐車が成功するまでこれを繰り返す。駐りに成功した場合、事前に定められたシーケンス数  $parking\_counter$  だけ駐車する。この駐車時間  $parking\_counter$  は 1 試行ごとに設定された範囲内 (1 時間 ~ 3 時間) の任意の時間である。駐車開始時に送信される位置情報には、駐車中を表すフラグが付与さ

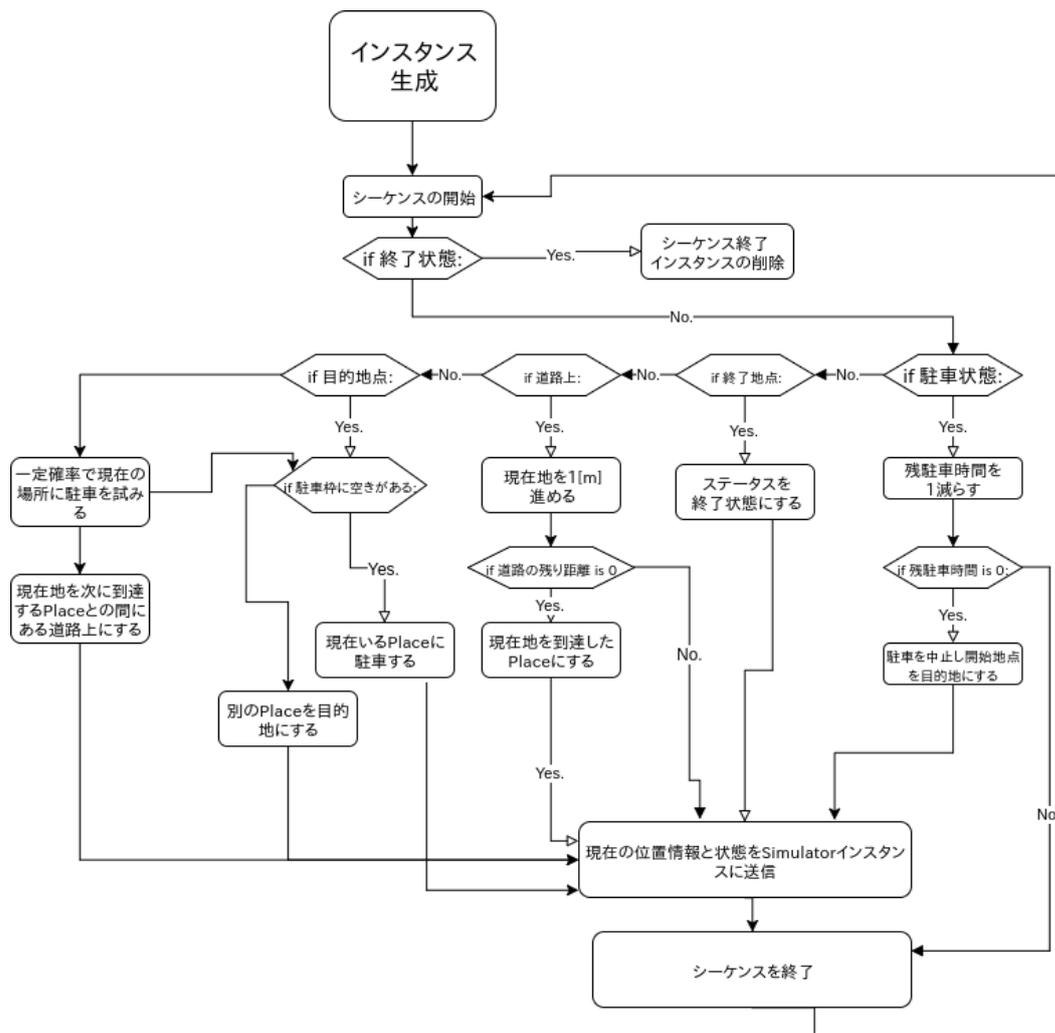


図 4.1: 1 シーケンス中の *Car* インスタンスの振る舞いを表す図

れる．なお，駐車中にはエンジンが停止しており機器の電源が切られていると想定し，*Car* インスタンスは位置情報の送信を行わない．駐車終了後，発進を表すフラグを付与した位置情報を送信した後に，次のシーケンスより開始地点を目的地にした移動を開始する．

- 退出

駐車終了後に開始地点に帰還した *Car* インスタンスは *Simulator* クラスから削除される．

### 位置情報の送信

*Car* インスタンスは1 シーケンスごとに位置情報を記録したメッセージを *Simulator* クラスに送信する．以下が *Car* インスタンス *C* が送信するメッセージ *M* のフォーマット

トである．

$$M = (I_C, S_C, G_C) \quad (4.1)$$

$$I_C = C \text{ の識別 } id \quad (4.2)$$

$$S_C = C \text{ の現在の状態} \quad (4.3)$$

$$G_C = C \text{ の位置情報} \quad (4.4)$$

ノード上に車両が存在している場合はそのノードの位置情報を，エッジ上に存在している場合はノード間の位置情報をエッジの重みで分割した値が送信される．なお，ノード間の距離は計算量の削減のためにユークリッド距離 [33] で求める．*Place* インスタンス  $P_a$  上の *Car* インスタンスの位置情報  $G_C$  は以下の式で表される．

$$G_C = g(x), x = G_a \quad (4.5)$$

$$g(x) = \text{誤差関数} \quad (4.6)$$

$$G_a = P_a \text{ の位置情報 (経度, 緯度)} \quad (4.7)$$

$$(4.8)$$

*Place* インスタンス  $P_a \cdot P_a$  間に位置するエッジ  $E$  に進入した *Car* インスタンスの位置情報  $G_E$  は以下の式で表される．

$$G_E = g(x), x = \frac{\sqrt{\sum (G_a - G_b)^2} \cdot t_n}{E_w} \quad (4.9)$$

$$g(x) = \text{誤差関数} \quad (4.10)$$

$$G_a = P_1 \text{ の位置情報 (経度, 緯度)} \quad (4.11)$$

$$G_b = P_2 \text{ の位置情報 (経度, 緯度)} \quad (4.12)$$

$$t_n = C \text{ が } E \text{ に進入してからの経過シーケンス} \quad (4.13)$$

$$E_w = E \text{ の重さ (距離)} \quad (4.14)$$

各 *Car* インスタンスが送信する位置情報には，実際の衛星測位システムの誤差 [17] を模した最大+10[m] 程度の誤差付与処理を行う．安田の調査 [34] によれば，位置情報測位システムによる誤差は正規分布のため，本実装の誤差付与処理では結果を正規分布させる．以下が本実装における誤差を付与する誤差関数  $g(x)$  を表す式である．

$$g(x) = x + \frac{R}{10^{4.5}} \quad (4.15)$$

$$x = \text{位置情報の真値} \quad (4.16)$$

$$R = \text{標準正規分布 (平均 0, 標準偏差 1) の乱数を返す関数} \quad (4.17)$$

予備実験として，誤差関数  $g(x)$  を 1000 回実行した．この時の位置情報の真値との距離の分布と密度を図 4.2 に示す．なお， $x$  は第 5 章にて評価実験を行う慶應義塾大学湘南藤沢キャンパスの座標を選択した．2 点間の距離の計算には GeoPy<sup>2</sup> を利用した．

<sup>2</sup>地球上の二組の座標から距離を割り出すことが出来るライブラリ．<https://github.com/geopy/geopy>

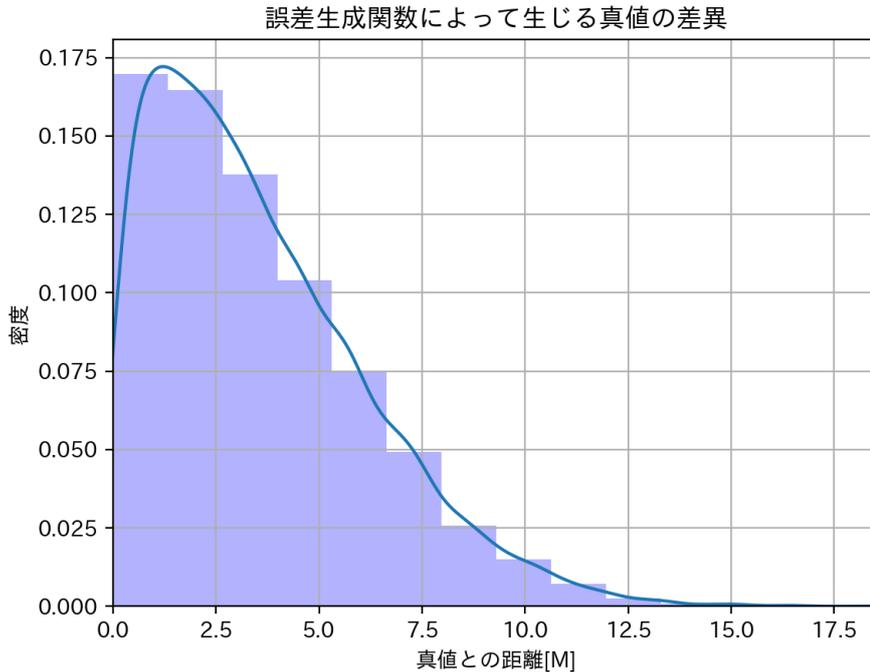


図 4.2: 誤差関数  $g(x)$  が返す位置情報と真値との差分の分布を表すヒストグラム

#### 4.1.4 時間経過と評価実験プログラムについて

本実装では, *Simulator* クラスが仮想地図と仮想車両を管理し, シミュレーションの進行を担う. 図 4.4 に主要なプロパティとメソッドを記述する.

class Simulator	
プロパティ名	概要
sequence	現在のシーケンスが挿入される int 型
map	Simulator インスタンスが保有する Map インスタンス
cars	Simulator インスタンスが保有する Car インスタンスが挿入される配列
car_pace	isgradulaly が True の場合に Car インスタンスを作成するペース float 型
isgradulaly	Car インスタンスの作成を徐々に行っていくかどうか bool 型
result	Car インスタンスから送信された位置情報を挿入する配列
メソッド名	概要
showPlot	Map インスタンスを可視化する
genCar	Car インスタンスを作成する際に実行される
next_sequence	シーケンスを進めるメソッド 各 Car インスタンスの next_seq メソッドを id. が若い順番に実行していく
クラスメソッド名	概要
execXmeans	XMeans インスタンスを作成しクラスターリングを実行する
divide_log	記録された Car インスタンスの位置情報群 (result) を各種別に分割する
find_near_place	引数として与えられた Place インスタンス群と座標からその座標に最も近い Place インスタンスを探す
infer_parking_slot	result を引数として受け取り, 駐車/発車ログから各駐車区画が保有する駐車枠の数を推測する
parking_place_test	本提案手法による駐車区画の数と位置情報の推測を実行するメソッド 成功の可否を返す
parking_slot_test	本提案手法による各駐車区画の保有する駐車枠の数の推測を実行するメソッド

表 4.4: 本シミュレータを表す *Simulator* クラス

以下に *Simulator* クラスが担う役割を列挙する.

1. *Map* インスタンス, *Place* インスタンスの作成  
事前に用意した JSON ファイル<sup>3</sup>に基づいて仮想地図を作成する。
2. *Car* インスタンスの作成  
実行引数の要素に則って仮想車両を作成する。
3. シーケンスの進行と位置情報ログデータの記録  
本実装では1単位時間を1シーケンスとして表現する。1シーケンスごとに各 *Car* インスタンスは移動し, *Simulator* インスタンスが位置情報ログデータを記録する。全ての *Car* インスタンスが駐車を完了し, 仮想地図から退出するまでシーケンスを進行する。
4. ログデータの変換  
すべての移動ログデータを, 移動 (*Move*) · 駐車 (*Parking*) · 発車 (*Departure*) の3つに分解する。
5. ネットワークモデルの推定処理の実行  
4.3 節にて後述するネットワークモデルの推定処理を実行する。
6. 推定結果の評価  
第5章にて後述する基準に則って推定結果の評価を行う。
7. 結果の可視化  
ネットワークモデル推定及びそれについての評価の結果の可視化を行う。

## 4.2 クラス図

図 4.3 は本実装の各クラスの関係を表した概念図である。 *Map* クラスと *Place* クラスは  $1:n$ , *Map* クラスと *Simulator* クラスは  $1:1$ , *Simulator* クラスと *Car* クラスは  $1:n$ , また *Place* クラスは *Car* クラスと  $1:n$  の関係性を持つ。

## 4.3 ネットワークモデルの推定

本実装では第3章で述べた提案手法を用いて, 得られた位置情報のログデータを解析し, 以下の3点についての自律的な推定を行う。

### 4.3.1 駐車区画座標の推定手法

本シミュレータ上の仮想車両が駐車をした際の座標群を, 石岡らによる提案手法 [31] を一部改良した分類手法に基づいてクラスタリングする。各クラスタの重心となる点を駐車区画の中心地点の座標と推定する。

<sup>3</sup>軽量データ記述フォーマットの一つ。 <https://www.w3.org/TR/json-ld/>

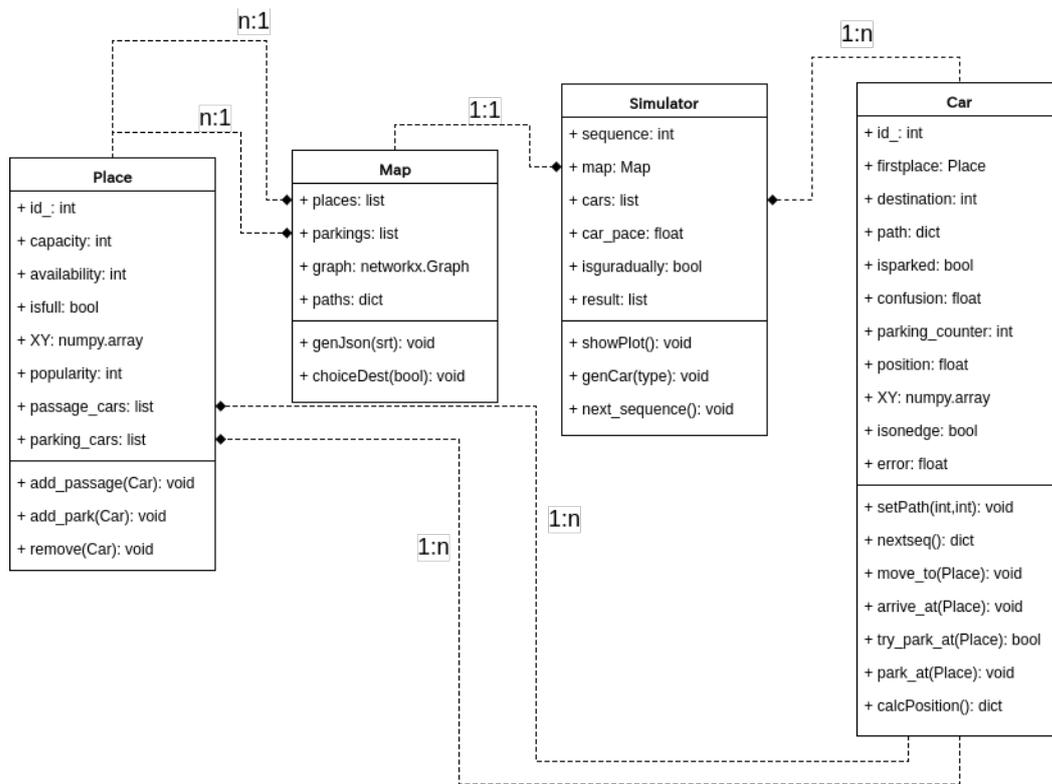


図 4.3: 本シミュレータの各クラスの関係性

## 用いる実装

すべての位置情報ログデータから *Park* フラグが付与されたログを抽出し, X-Means 法 [30] を用いてクラスタリングする. 本実装では石岡らによる手法 [31] を Python<sup>4</sup> 環境に対応させた Yuichi Goto(yasaichi) 氏の実装 [35][36] を一部改良して用いる. 改良点は以下の2点である. これによりクラスタリング中の例外発生を防止することが出来る.

- 共分散行列の利用

Goto 氏による実装 [36] では BIC(ベイズ情報量基準) を求める際の指標として, 不偏共分散を利用している.  $n$  組のデータ  $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$  が存在するとき, 不偏共分散  $s_{xy}$  は以下のように示すことが出来る.

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (4.18)$$

不偏共分散は母集団全てを計算に用いることが出来ない場合に有効性を発揮する指数であるが [37], 本手法ではそのクラスター全てのデータを算出に利用するため, 必ずしも不偏共分散を利用する必要がない. また, 本環境では再帰的にクラスタリングを実行する際に, クラスタ内のデータの個数  $n$  が 1 になる場合がある.

<sup>4</sup>オブジェクト指向プログラミング言語とその実行環境. <https://www.python.org/>

そのため、本実装では以下の式で表される共分散  $s'_{xy}$  を用いる。

$$s'_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (4.19)$$

- 尤度を求める際に特異共分散行列を許可する  
前述の通り、本環境ではクラスタリングを実行する際にクラスタ内のデータの個数  $n$  が 1 になる場合がある。この時、共分散行列  $s'_{xy}$  は特異行列となる。本実装で用いた対数尤度関数<sup>5</sup> はデフォルトで特異行列を許可しないためプログラムが異常終了する。そのため、本実装では対数尤度関数に特異共分散行列を許可する引数を与える。

### 4.3.2 駐車区画の有する駐車枠数の推定手法

すべての位置情報ログデータから、*Park* フラグと *Departure* フラグが付与されたログを抽出し、各駐車区画の最大駐車台数を計数し、その区画の有する駐車枠数とする。サンプルとした時間  $\max\{t\}$  中にこの駐車区画が満車状態になる場合にのみ、正しい駐車枠数を推定することが出来る。

#### 用いる実装

本実装では *Simulator* クラスのクラスメソッドに *infer\_parking\_slot* 関数を用意した。以下に手順の概略を示す。

1. 引数を与える。  
すべての位置情報ログデータが挿入されている配列 *all\_result* と、第 4.3.1 項で求められた駐車区画の座標群 *parking\_positions* を引数として与える。
2. 推測された駐車区画群を示す配列 *parkings* を作成する。  
要素には駐車区画を表す辞書型のオブジェクトを挿入する。この辞書は仮識別子 *id\_*、座標 *position*、駐車枠数 *capacity*、駐車している車両数 *cars* のキーを有する。*capacity*、*cars* には初期値 0 を挿入する。
3. 位置情報ログデータの走査  
すべての位置情報ログデータ *all\_result* を走査する。*Parking* フラグを確認した際には該当する駐車区画の辞書の要素に対して  $cars = cars + 1$  を行い、*Departure* フラグが確認された場合には  $cars = cars - 1$  を行う。

<sup>5</sup>Python 環境で動作する数値計算ライブラリ SciPy の `stats.multivariate_normal` 関数 [https://docscipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.multivariate\\_normal.html](https://docscipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.multivariate_normal.html)

4. *cars* 要素の最大値を *capacity* に代入する .  
 前述の走査中に合わせて *cars* の最大値を記録する . すべての走査が終了した際に *cars* の最大値を *capacity* に代入する .

### 4.3.3 駐車区画・交差点間を結ぶ道路の推定手法

Edelkamp 氏が提案した k-Means 法をベースとした手法 [23] を用いることを想定する . 全ての位置情報ログから駐車時以外のログを抜粋した位置情報群を X-Means 法にてクラスタリングした際の各クラスターの中心点を結びフィッティングを行う . 各エッジの重みに関しては位置情報ログから車両の移動にかかるシーケンスを算出し , 統計的危険値を取り除いた後の平均値とする .

なお , 本論文では Biagino 氏らの評価実験 [21] によって有効性が示されているため , 評価実験の対象から除外する .

## 4.4 実装環境

本説では本シミュレータの実行環境について記述する .

本シミュレータ及び推定システムは Python<sup>6</sup> とそのライブラリ群を用いて実装した .

図 4.4 は第章で記述する評価実験を実行した環境を表している .

実行環境	
ソフトウェア	バージョン
Python	3.6.2(Anaconda3-5.0.0)
NumPy <sup>5</sup>	1.13.1
NetworkX <sup>6</sup>	1.11
GeoPy <sup>7</sup>	1.11.0
SciPy <sup>8</sup>	8.99
matplotlib <sup>9</sup>	2.0.2
setproctitle <sup>10</sup>	1.11.10
XMeans	<a href="https://gist.github.com/yasaichi/254a060eff56a3b3b858">https://gist.github.com/yasaichi/254a060eff56a3b3b858</a>
ハードウェア	チップセット
HP ProLiant DL320e Gen8 v2 Intel(R) Xeon(R) CPU E3-1240 v3 @ 3.40GHz	

表 4.5: シミュレータの実行環境

<sup>6</sup>2018年1月現在 , Python 言語の現行仕様バージョンは Python3 と呼ばれる . <https://www.python.org/>

- 
- <sup>7</sup> 数学的な計算や行列式に用いるライブラリ . <http://www.numpy.org/>
- <sup>8</sup> ネットワークグラフの構築や描写, 最短経路の計算などに用いるライブラリ . <https://networkx.github.io/>
- <sup>9</sup> 地球上の座標を扱うためのライブラリ . 本実装では座標間の距離計算に用いた . <https://github.com/geopy/geopy>
- <sup>10</sup> 数値計算用ライブラリ . <https://www.scipy.org/>
- <sup>11</sup> ネットワークグラフや図表のプロットを行う . <https://matplotlib.org/>
- <sup>12</sup> 実行中のプロセス名を変更するライブラリ . <https://github.com/dvarrazzo/py-setproctitle>

## 第5章 評価

第4章にて述べたシミュレータを用いて，本提案手法の妥当性を検証するための評価実験を行う．本章では評価実験の概要と，それにより得られた結果及び考察について述べる．

### 5.1 仮定

本節に本評価実験の仮定条件となる事項を記述する．

#### 5.1.1 運用環境のモデルケース

3.3節で述べたように，本研究の提案手法では以下のような環境の駐車場を想定している．

- 駐車場全体が広く，区画の空き状況を利用者が一度に確認することが困難である．
- 駐車区画の人気度に偏りがある．
- 運営者による満空情報の開示が難しい．

本論文ではそれらに当てはまるモデルケースとして，慶應義塾大学湘南藤沢キャンパス(以後SFC)の各駐車区画を本シミュレータ上に再現した．各駐車区画の座標，座標間の距離はGoogleMaps[38]より参照した．図5.1にGoogleMaps上にプロットしたSFCのネットワークモデルを示す．評価実験を明確にするために一部を簡略化している．

図5.2が本シミュレーター上に再現したSFCのネットワークモデルである．駐車区画を $P$ ，その他の道路との接続点を $C$ としている．車両が駐車場内に進入する箇所を0として，それぞれ19まで識別コードを割り当てる．

#### 各ネットワークノードの特徴

本評価実験において，各ネットワークノードの駐車枠数や特徴，各駐車場の人気度などは以下の表5.1の通り指定する．なお，各駐車区画の有する駐車枠数は2018年1月に実施した予備調査で計数したもので，人気度は任意に設定したものである．



図 5.1: googleMaps 上にプロットした SFC 内駐車区画のネットワークモデル

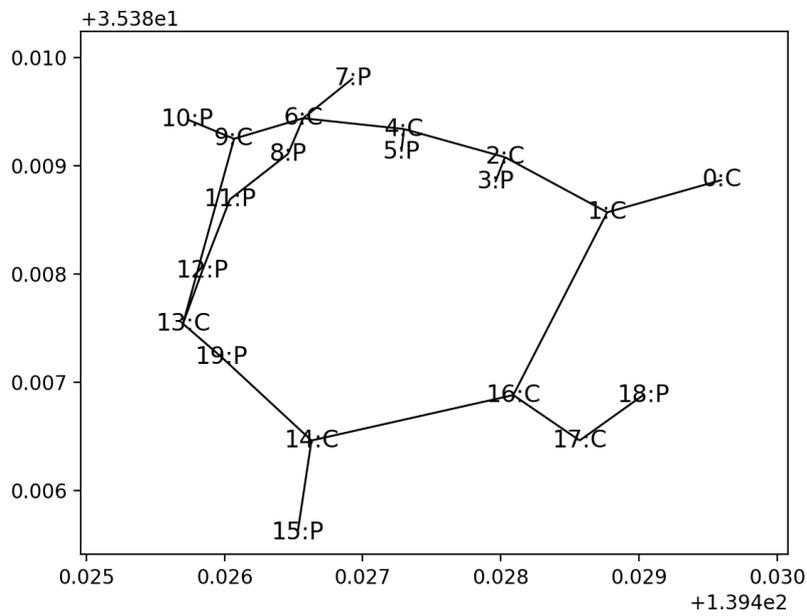


図 5.2: ネットワークモデル化した SFC

本評価実験における各ネットワークノードの特徴						
Place ID	駐車場	駐車枠数	人気度	座標 [緯度, 経度]	隣接するノードと距離 [m]	備考
0	×	-	-	35.38887, 139.4296	1:82	スタート地点
1	×	-	-	35.38857, 139.42877	0:82, 1:82	
2	×	-	-	35.38908, 139.42803	1:82, 3:26, 4:72	
3		14	10	35.38885, 139.42796	2:26	アルファ館横
4	×	-	-	35.38934, 139.4273	2:72, 5:23, 6:67	
5		16	5	35.38913, 139.42728	4:23	シータ館・メディア館
6	×	-	-	35.38944, 139.42657	4:67, 7:61, 8:39, 9:50	
7		80	5	35.38981, 139.42693	6:61	体育館前
8		7	20	35.38911, 139.42646	6:39, 11:61	ラムダ館横
9	×	-	-	35.38925, 139.42607	6:50, 10:37, 13:197	
10		16	15	35.38943, 139.42573	9:37	タウ館横
11		14	25	35.38869, 139.42604	8:61, 12:76	イオタ館, オミクロン館
12		14	10	35.38803, 139.42584	11:76, 13:56	カッパ館, イプシロン館
13	×	-	-	35.38754, 139.4257	9:137, 12:56, 19:43	
14	×	-	-	35.38646, 139.42663	15:95, 16:140, 19:104	
15		14	30	35.38561, 139.4265	14:95	南門警備室前
16	×	-	-	35.38688, 139.42809	1:198, 14:140, 17:94	
17	×	-	-	35.38646, 139.42857	16:64, 18:67	
18		23	15	35.38688, 139.42903	17:67	テニスコート前
19		24	5	35.38723, 139.42598	13:43, 14:104	生協棟前

表 5.1: SFC の仮想地図内における各区画の属性

## 5.1.2 条件設定

本評価実験の際に設定する条件について記述する。

- 1シーケンスの時間  
本実装では1シーケンスを1秒 ( $t = 1[\text{秒}]$ ) とする。これは利用車両の位置情報送信間隔を1秒と定義することに等しい。
- 各仮想車両の目的地の決定  
本シミュレータでは車両が目的の区画で駐車が出来ない場合、他の区画を次の目的地として設定して場内を巡回する。その時、車両は通過した駐車区画に関してその区画の満空状況を確認出来るものとする。また、目的地ではない場合も車両ごとに定められた確率 ( $Car$  クラスの  $confuse$  プロパティ) に依って駐車を試みる。
- 仮想車両の駐車時間  
本評価実験では車両の駐車時間  $p_t$  は、1時間から3時間 ( $3600[\text{秒}] < p_t < 10800[\text{秒}]$ ) までの任意の時間とする。
- 各駐車区画の人気度  
各駐車区画の人気度は表 5.1 で定めたものとする。
- 道路・交差点の交通量のキャパシティ  
本評価実験では各道路に収容できる交通量を制限しないものとする。仮想車両群

は各駐車区画を無制限に通過することが出来る。一方で、区画が保有する駐車枠に空きが無い場合は、駐車を行わず通過するものとする。

- 位置情報ログの生成  
多様なパターンに対応するため、試行の度にシミュレータを実行し、仮想車両群の位置情報ログデータを生成する。
- 試行回数  
本評価実験では、変数を変える度にシミュレータの実行・ネットワークモデルの推定を各説明変数につき 100 回ずつ行う。
- 座標の誤差  
車両から送信される位置情報に対して第 4.1.3 項で述べた誤差付与処理を施す。生成される誤差の分布に関しては図 4.2 を参照されたい。

## 5.2 評価実験

本提案手法の妥当性を検証するために 2 つの評価実験を行った。本節ではその概要と結果及び考察について述べる。

### 5.2.1 評価実験 1：駐車区画の位置推定

本項では、第 3.4.2 節で述べた駐車区画の位置推定手法を、定量的に評価するための評価実験 1 について記述する。

#### 条件と変数

評価実験 1 では第 5.1.2 節に挙げた条件の他に、以下のような条件を設定する。

- 仮想車両のスタート方法  
すべての車両を同時にスタート地点から仮想地図内に進入させる。
- 最大シーケンス数 (収集対象時間)  
評価実験 1 では最大シーケンス数を設定しない。すべての車両が駐車を完了し SFC から退出するまでの位置情報ログを推定に用いる。
- 車両の数  
5.2.1 項に後述するように、評価実験 1 では仮想地図内に進入する車両の台数を説明変数とする。本実験での車両台数は、実環境下での利用車両の述べ台数とみなすことが出来る。

## 評価項目

評価実験 1 では、以下の 2 項目に関して順に推定結果を検証し、推定の成功の可否を検証する。

### 1. 駐車区画の数

システムが推定した駐車区画の数と、真値の駐車区画の数が一致するかどうかを検証する。一致した場合次の項目に進む。

### 2. 駐車区画の座標

システムが推定した各駐車区画の座標群と、真値の座標群との距離が真値の駐車区画間の最短距離より小さい場合、推定が成功したと判断する。

## 説明変数と目的変数

評価実験 1 では説明変数  $x$  を仮想地図内に進入する台数とし、推定結果の成功率  $y$  を目的変数とする。 $n$  回試行した時の成功率  $y$  は以下の式のように表される。

なお、本システムの推定機構  $f(x)$  は推定が成功した場合 1 を、失敗した場合 0 を返す。また、本実験では  $x = 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600$  ,  $n = 100$  とする。

$$y = \frac{\sum_{i=1}^n f(x)}{n}, x = \text{車両の台数} \quad (5.1)$$

$$f(x) = \text{本システムの推定機構} \quad (5.2)$$

$$n = \text{総試行回数} \quad (5.3)$$

## 結果

$x = 250$  の一度の試行を例に、試行が成功した場合 ( $f(x) = 1$ ) に、システムが推定した駐車区画の座標とクラスタリング処理を施した駐車時の座標群をプロットした結果を図 5.3 に示す。青い円は真値の駐車区画座標、線群は真値のネットワークモデルの道路を表すエッジである。

評価実験 1 の結果を図 5.4 に示す。 $x > 450$  の場合に成功率が 90% を超えていることが読み取れる。

## 考察

第 5.1.1 節より、本実験が対象とした SFC キャンパスの総駐車枠数は 222[台] であるため、90% 以上の成功率を得るためには、総駐車枠の 200% 以上の述べ車両台数が必要であることがわかった。1 日のうちに駐車区画の多くが満車近くになる SFC の駐車場の場合、データ推定対象となる時間を 2 日以上に設定することで、十分に信頼できるネットワークモデル推定を行うことが出来ると言える。

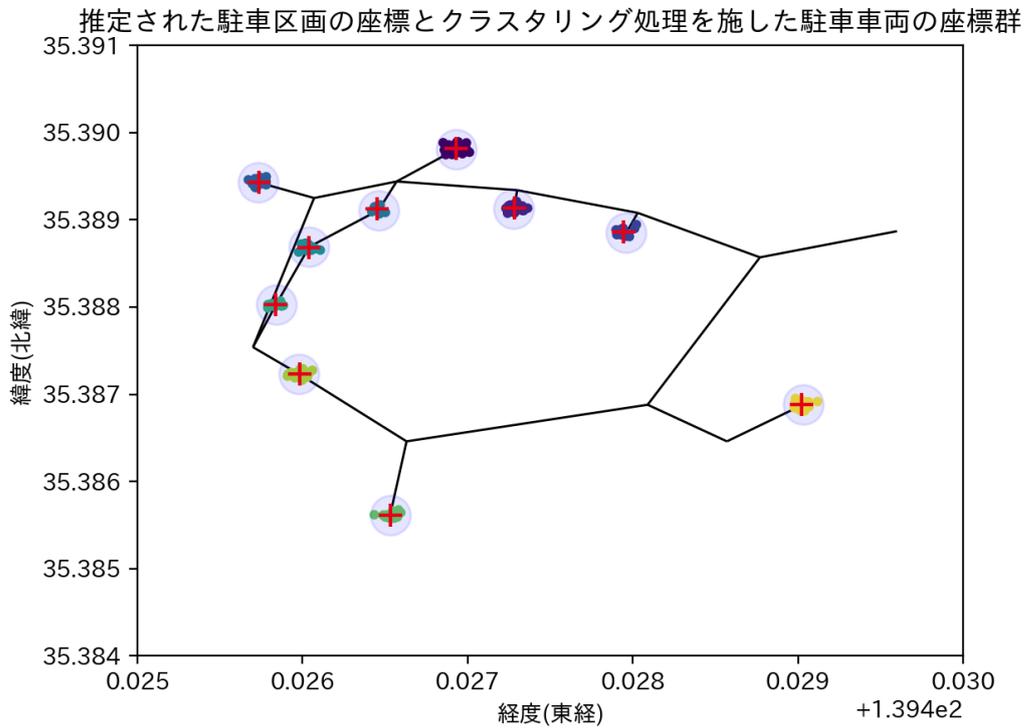


図 5.3: 評価実験 1 が成功したケースにおける推定された駐車区画座標と真値の比較

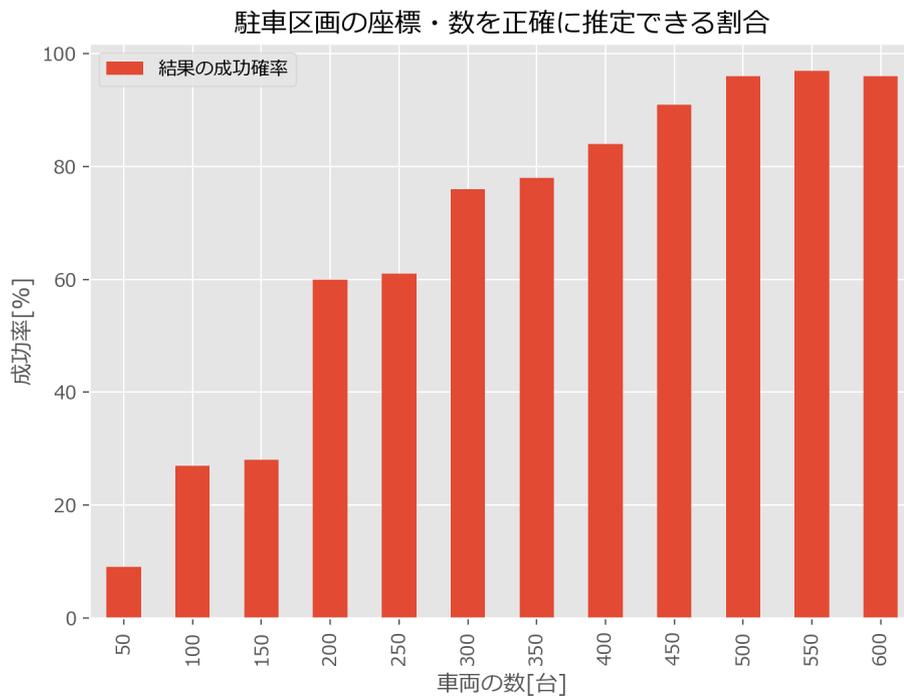


図 5.4: 利用車両の台数と推定の成功率の関係性

## 5.2.2 評価実験 2：駐車区画が保有する駐車枠数の推定

本項では、第 3.4.3 節で述べた、駐車区画が保有する駐車枠数の推定手法を定量的に評価するための評価実験について記述する。

### 条件と変数

評価実験 2 では、第 5.1.2 節に挙げた条件の他に、以下のような条件を設定する。

- 仮想車両のスタート方法  
すべての車両を同時にスタート地点から進入させる。
- 最大シーケンス数 (収集対象時間)  
評価実験 2 では最大シーケンス数を設定しない。すべての車両が駐車を完了し、SFC から退出するまでの位置情報ログを本評価実験での推定に用いる。
- 車両台数  
5.2.1 項で述べたように、評価実験 1 では、導入する車両台数に最大シーケンス数中の述べ車両台数という意味付けを行った。一方で、評価実験 2 では場内に存在する最大車両台数と換言する。
- 前提となる推定結果  
駐車枠数の推定を行うためには駐車区画の数・座標の 2 項目が正しく推定されていることが必要である。よって、第 5.2.1 項で定めた基準を満たした場合の推定結果を前提条件とする。

### 評価項目

評価実験 2 では、以下の項目に関して推定結果を検証し、推定の妥当性を検証する。

#### 1. 充足率

駐車区画  $p$  に関して、システムが推定した保有枠数  $a_p$  の真値  $G_p$  に対する割合を推定充足率  $F_p = \frac{a_p}{G_p}$  と定義する。  $F_p = 1$  の時、真値と同じ値を推定したと見なすことが出来る。なお、3.4.3 項より、 $a_p \leq G_p$  であるため、 $F_p \leq 1$  である。

### 説明変数と目的変数

評価実験 2 では、仮想地図内に進入する車両の台数を説明変数  $x$  とし、 $n$  回試行した際の、駐車区画  $p$  の推定充足率  $F_{pn}$  の中央値  $\tilde{F}_{pn}$  を目的変数  $y_p$  とする。なお、本実験

において,  $x = 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600$ , 試行回数  $n = 100$  とする.

$$y_p = \tilde{F}_{pn} \quad (5.4)$$

$$F_{pn} = n \text{ 回試行した際の推定充足率群} \quad (5.5)$$

$$F_p = \frac{a_p}{G_p} \quad (5.6)$$

$$f_2(x, p) = a_p \quad (5.7)$$

$$f_2(x, p) = \text{本推定手法, } x = \text{車両の台数, } p = \text{駐車区画} \quad (5.8)$$

$$n = \text{総試行回数} \quad (5.9)$$

## 結果

本項では評価実験 2 の結果について述べる．推定された推定充足率の中央値  $\tilde{F}_{pn}$  と, 車両数  $x$  との関係を図 5.5 に示す．図 5.6 に各駐車枠ごとの結果を示す．

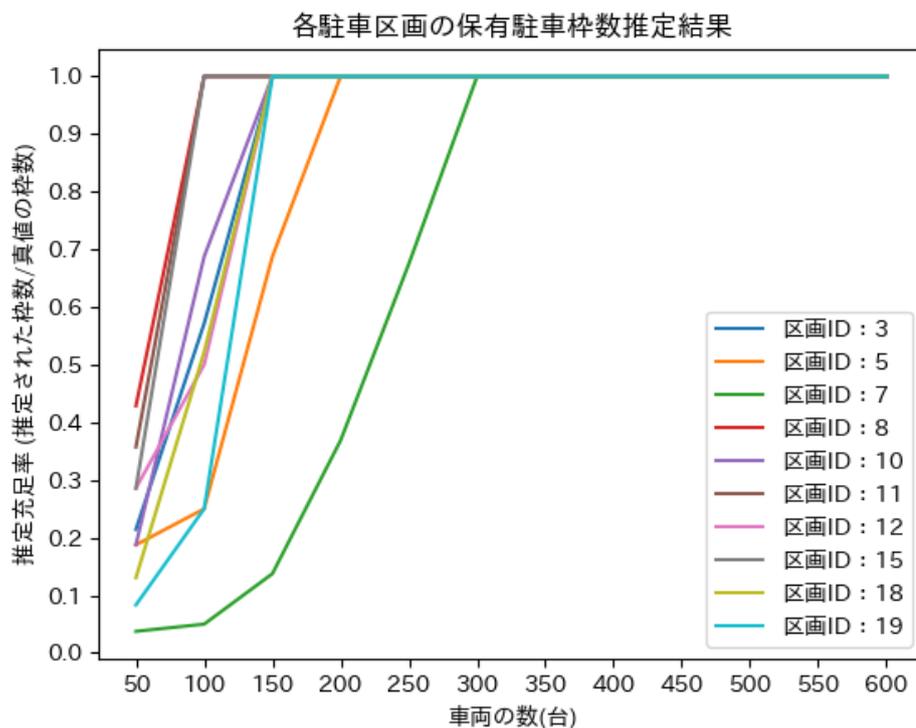


図 5.5: 仮想地図内の最大車両台数と各駐車区画の推定充足率

## 各駐車区画の推定充足率分布の比較

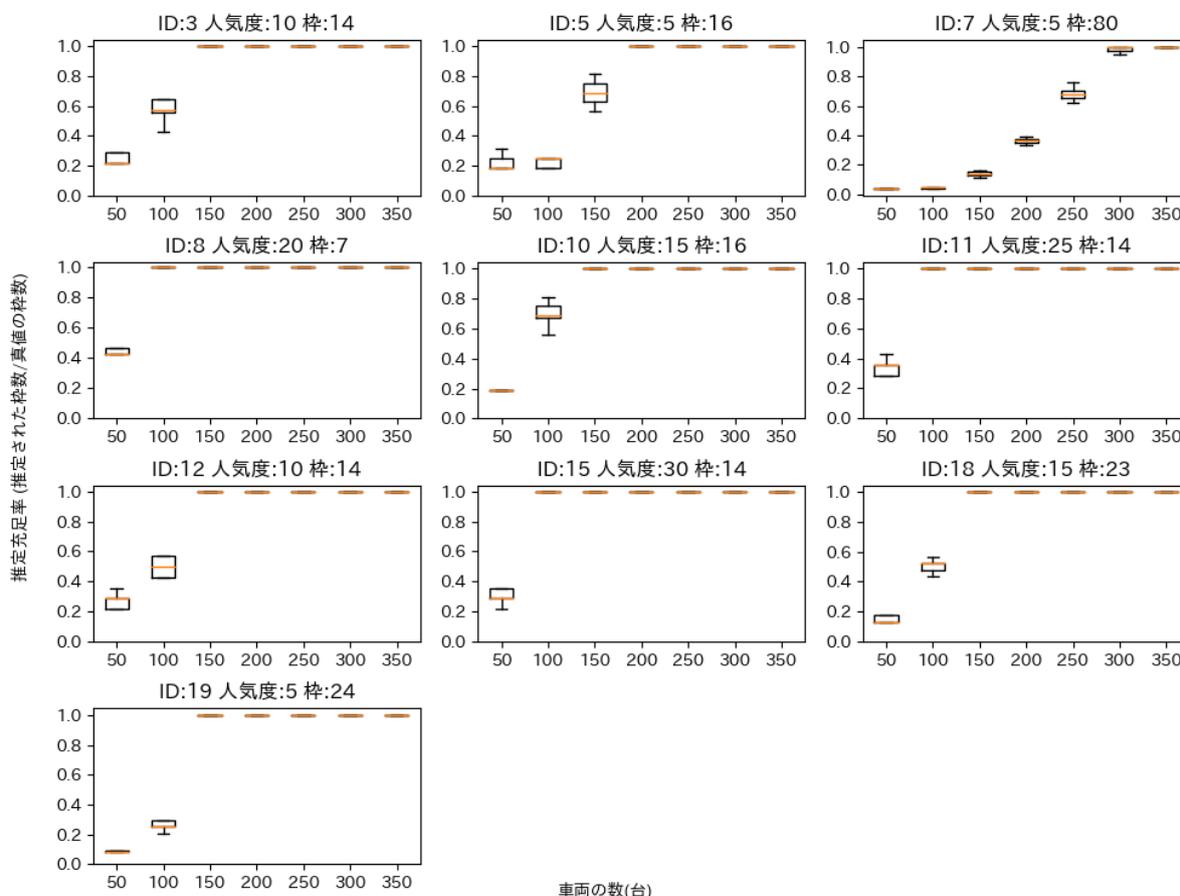


図 5.6: 仮想地図内の最大車両台数と各駐車区画の推定充足率

### 考察

図 5.5 より、多くの駐車区画が  $x > 200$  の時、駐車枠数の完全な推定に成功しているケースが多いことが読み取れる。また、人気度の低い2つの区画を除けば、 $x > 150$  の時完全な推測に成功する可能性が高いことがわかる。第 5.1.1 節より、本実験が対象とした SFC キャンパスの総駐車枠数は 222[台] であるため、駐車場内の満車率が 70% 程度になった場合に、主要な駐車区画の保有する枠数を推定できたと言える。

また、図 5.6 より、区画 ID:15 や ID:11 のような特に人気が高い駐車区画に関しては、 $x > 100$  の場合に推測に成功していると言える。人気度が特に高い駐車枠に限れば、満車率が 50% 程度あれば枠数の把握が可能になることが導かれる。

しかしながら、区画 ID:7 では、駐車場の総枠数 (222[台]) 以上に車両が場内に進入していても推定充足率が 1 に満たない。これは、空き枠のある駐車区画があるのにも関わらず、駐車場内を巡回している車両が多く発生していることを意味している。この現象は、各駐車区画の満空状況を各車両が走行中に把握することが出来ない (5.1.2 節を参照) ことが原因であると考えられる。本評価実験の結果より、人気度が低く枠数が

多い駐車区画では、現実的な車両台数では推測できないことが明らかになった。

## 第6章 結論

本章では本論文のまとめと今後解決すべき課題を示す。

### 6.1 本研究のまとめ

本論文では、始めに、路上に敷設した機器を用いる既存の中央管理型駐車場管理システムや、車々間通信による自律分散型管理システムに関する問題点を整理し、その代表として、ネットワークモデルの構築とシステムの管理が運営者の大きな負担であり導入障壁であると指摘した。

次に、既存の研究や手法から、自動車の位置情報を用いる駐車場の最適化手法の有用性を示した。

そこで本研究では、路上に設置する機器や運営主体による管理に依存せずに駐車区画の空き状況を収集・可視化出来る環境を創出することを目的として、自動車の位置情報ログデータによって駐車場のネットワークモデルを自律的に推定する手法と、得られたネットワークモデルによって自律的に駐車場の満空状況を管理する新しいアーキテクチャを提案した。

また、駐車場のネットワークモデルを構成する要素を整理し、駐車区画の位置座標と保有する駐車枠数の推定を取り組むべき課題に設定した。本論文では、車両の駐車ログデータと X-Means 法を用いたアプローチと、駐車区画の人気度の偏りに注目したアプローチによって自律的に推定する手法を提案した。

加えて、本提案手法の妥当性を検証するために、慶應義塾大学湘南藤沢キャンパスをモデルケースとしたシミュレータを実装し、2つの評価実験を行った。駐車区画の座標に関しては、駐車場の述べ200%以上のログデータを収集・解析することで、十分に信頼できる推定が行えることを示した。また、主要な駐車区画の駐車枠数に関しては、駐車場全体の満車率が70%程度のログデータを用いることで、高い精度の推定が行えることを示した。

以上の実験の成果として、シミュレータ環境における本提案手法の妥当性が証明された。

### 6.2 将来的な展望

本論文では妥当性の検証方法として実地図をモデルにしたシミュレータ環境を用いたが、実環境下における有用性は再度検証する余地があると言える。

今後は、実際の車両の走行ログデータや、車載器の低普及環境下での検証を引き続き行い、本提案手法のフィジビリティを明確にしていくことが求められる。

本提案手法の推定精度を更に高めることで、実環境下においても路上常設のセンサーを用いることなく駐車場全体の空き状況の把握・可視化・最適な区画へのナビゲーションを行うことが可能になり、利用車両の無駄な徘徊の低減や駐車時間の減少を見込める。加えて、都市部や観光地の様な同様の問題が発生しやすい環境下においても、駐車場運営主体の枠を超えた空き状況の統合的な可視化の実現と、渋滞の低減に対する取り組みが期待される。

# 謝辞

本論文の執筆にあたり，ご指導いただきました慶應義塾大学政策メディア研究科委員長 村井純博士，環境情報学部教授 楠本博之博士，同学部教授 中村修博士，同学部教授 Rodny D. Van Meter 博士，環境情報学部教授 三次仁博士，同学部准教授 植原啓介博士，政策・メディア研究科特任准教授 佐藤雅明博士，同研究科特任准教授 鈴木茂哉博士，環境情報学部講師 斉藤賢爾博士，政策・メディア研究科特任助教 工藤紀篤博士，同研究科特任助教 空閑洋平博士，同研究科特任講師 松谷健史博士，同元特別招聘教授 下村健一氏，同研究科永山翔太博士に感謝致します。特に二年次より，研究，外部との共同プロジェクト，学業，進路，私生活の多方面において専らのご指導を賜った佐藤雅明博士，四年次より研究，研究室ネットワーク管理に関してご指導下さいました植原啓介博士に深謝の意を表します。“インターネット自動車”という興味深い研究テーマを私にご教示戴きました両氏にただただ感謝の限りです。また，遠くハンガリーの地から私の中間発表ポスターにアドバイスを下さいました永山翔太博士，研究のあり方や私生活に関して貴重なアドバイスを下さいました工藤紀篤博士に心よりお礼申し上げます。

また，インターネット自動車研究グループのメンバーとして共に切磋琢磨した，総合政策学部 須田小百合氏，西田亘氏，前田樹氏，環境情報学部 神智尚氏，鈴木雄祐氏，広田和也氏，山田航太郎氏，浅沼沙理氏に深く感謝いたします。特に英語に疎い私を集中特訓してくれた須田小百合氏には感謝してもしきれません。加えて，総合政策学部 深川祐太氏，環境情報学部鈴木雄祐氏，熊谷啓孝氏と取り組んだ外部との研究プロジェクトは私の大きな経験値になりました。諸氏に大変感謝致します。

そして村井研究室での研究，日常生活を共にした優秀な学生の皆様に大変感謝致します。特に，同期の尾崎周也氏，東海林晃氏，河口綾摩氏，桑原誠尚氏，亀井大向氏，神智尚氏，鎧坂文菜氏のような締め切り直前まで生活を共に闘いぬいた「戦友」の存在に日々励まされました。また，先輩として私の研究活動を特にご指導戴きました鈴木恒平氏，加藤桂子氏，阿部涼介氏，小林茉莉子氏，澤井優作氏，大戸浩司氏には特段の感謝の限りです。

最後に，生活，精神両面で多大なる応援をして下さった父 豊田力生氏，母 恭子氏，姉 晏以氏，妹 安莉氏，犬 りんな氏に厚くお礼申し上げます。家族のサポートあってこそ，ここまで生きてこれたといっても過言ではありません。

以上を以って本論文の謝辞とさせていただきます。私を支えて下さった全ての方に重ねて感謝申し上げます。

## 参考文献

- [1] 国土交通省. 観光地における渋滞対策, oct 2016. <http://www.mlit.go.jp/common/001150187.pdf> 最終確認日 2017年12月19日.
- [2] 山崎 瑠香, 佐々木 拓馬, and 花泉 弘. ステレオカメラを用いる空き駐車スペース検出法. 自動制御連合講演会講演論文集, 59:14–17, 2016.
- [3] タイムズ駐車場検索. <https://times-info.net/> 最終確認日 2017年12月19日.
- [4] Elias C. Eze, Sijing Zhang, Enjie Liu, and Joy C. Eze. Advances in vehicular ad-hoc networks (vanets): challenges and road-map for future development. *International Journal of Automation and Computing*. Recent advances in wireless communication technologies and auto-mobile industry have triggered a significant research interest in the field of vehicular ad-hoc networks (VANETs) over the past few years. A vehicular network consists of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications supported by wireless access technologies such as IEEE 802.11p. This innovation in wireless communication has been envisaged to improve road safety and motor traffic efficiency in near future through the development of intelligent transportation system (ITS). Hence, governments, auto-mobile industries and academia are heavily partnering through several ongoing research projects to establish standards for VANETs. The typical set of VANET application areas, such as vehicle collision warning and traffic information dissemination have made VANET an interesting field of mobile wireless communication. This paper provides an overview on current research state, challenges, potentials of VANETs as well as the ways forward to achieving the long awaited ITS.
- [5] 国土地理院. Gnss を使用した測量のいろいろ. [http://terras.gsi.go.jp/geo\\_info/GNSS\\_iroiro.html](http://terras.gsi.go.jp/geo_info/GNSS_iroiro.html) 最終確認日 2017年12月19日.
- [6] 山下翔平 and 高見一正. 大規模駐車場での区画管理クラスタ間の車車間オーバレイトポロジー構成法と空塞情報収集法. In 第76回全国大会講演論文集, volume 2014, pages 127–128, mar 2014.
- [7] D. Jiang and L. Delgrossi. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2036–2040, May 2008.

- [8] 総務省 東海総合通信局. Dsrc (狭域通信) システム. <http://www.soumu.go.jp/soutsu/tokai/musen/its/dsrc.html> 最終確認日 2018 年 1 月 17 日.
- [9] 社団法人 電波産業会. Arib std-t75 1.5 版, Sep 2008. [https://www.arib.or.jp/kikaku/kikaku\\_tushin/desc/std-t75.html](https://www.arib.or.jp/kikaku/kikaku_tushin/desc/std-t75.html).
- [10] 株式会社デプロ. Ieee 802.11p. <http://www.de-pro.co.jp/2016/11/22/8959/> 最終確認日 2017 年 12 月 19 日.
- [11] ITU. Itu-d technologies infrastructures and applications: Mobile communications. [https://www.itu.int/ITU-D/tech/MobileCommunications/IMT\\_INTRODUCING/IMT\\_2G3G4G.html](https://www.itu.int/ITU-D/tech/MobileCommunications/IMT_INTRODUCING/IMT_2G3G4G.html) 最終確認日 2017 年 12 月 19 日.
- [12] 杉野 勲. 2020 年に向けた 5 g 及び its・自動走行に関する総務省の取組等について, Jun 2017. <https://www.kiai.gr.jp/jigyuu/h29/PDF/0608p1.pdf> 最終確認日 2017 年 12 月 19 日.
- [13] Iso/tc 204 intelligent transport systems. <https://www.iso.org/committee/54706.html> 最終確認日 2017 年 12 月 19 日.
- [14] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys Tutorials*, 13(4):584–616, Fourth 2011.
- [15] R. Lu, X. Lin, H. Zhu, and X. Shen. Spark: A new vanet-based smart parking scheme for large parking lots. In *IEEE INFOCOM 2009*, pages 1413–1421, April 2009.
- [16] 孫為華, 剣持真弘, 柴田直樹, 安本慶一, and 伊藤実. 待ち時間を短縮する駐車ナビゲーション. In *マルチメディア通信と分散処理ワークショップ論文集*, volume 2013, pages 9–19, nov 2013.
- [17] GPS.gov. Gps accuracy. <https://www.gps.gov/systems/gps/performance/accuracy/> 最終確認日 2017 年 12 月 19 日.
- [18] 内閣府. 衛星測位サービス | みちびきについて | みちびき (準天頂衛星システム: qzss) 公式サイト. [http://qzss.go.jp/overview/services/sv04\\_pnt.html](http://qzss.go.jp/overview/services/sv04_pnt.html) 最終確認日 2017 年 12 月 19 日.
- [19] Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. Parknet: Drive-by sensing of road-side parking statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 123–136, New York, NY, USA, 2010. ACM.

- [20] V. Coric and M. Gruteser. Crowdsensing maps of on-street parking spaces. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 115–122, May 2013.
- [21] James Biagioni and Jakob Eriksson. Inferring road maps from gps traces: Survey and comparative evaluation. In *In Transportation Research Board, 91st Annual*, 2012.
- [22] E. W. FORGY. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, 21:768–769, 1965.
- [23] Stefan Edelkamp and Stefan Schrödl. Computer science in perspective. chapter Route Planning and Map Inference with Global Positioning Traces, pages 128–151. Springer-Verlag New York, Inc., New York, NY, USA, 2003.
- [24] Lili Cao and John Krumm. From gps traces to a routable road map. pages 3–12, November 2009.
- [25] 植木 優夫 and 笛田 薫. カーネル密度推定におけるカーネル関数の比較. 日本計算機統計学会大会論文集, 17:147–150, 2003.
- [26] J. J. Davies, A. R. Beresford, and A. Hopper. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5(4):47–54, Oct 2006.
- [27] James Biagioni. Inferring Road Maps from GPS Traces. <https://www.youtube.com/watch?v=byJzKjCta3c> 最終確認日 2017 年 12 月 19 日.
- [28] 豊田安信, 佐藤雅明, and 村井純. W3C Vehicle API を用いた複数のアプリケーションからの車両情報取得技術とプロトコル. In *マルチメディア, 分散協調とモバイルシンポジウム 2017 論文集*, volume 2017, Jul 2017.
- [29] 芦村 和幸. Web をクルマにつなぐ w3c の国際標準化- w3c vehicle api 標準化動向及び仕様草案の概要, 2016.
- [30] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00*, pages 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [31] 石岡 恒憲. クラスタ数を自動決定する k-means アルゴリズムの拡張について. *応用統計学*, 29(3):141–149, 2000.
- [32] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271, December 1959.

- [33] TIBCO Software Inc. How To Group Objects Into Similar Categories, Cluster Analysis. <http://www.statsoft.com/Textbook/Cluster-Analysis> 最終確認日 2018 年 1 月 15 日.
- [34] 安田 明生. Gps の概要. 32(11):646–653, 2003.
- [35] Yuichi Goto. Python で x-means 法を実装した. <http://web-salad.hateblo.jp/entry/2014/07/19/200347>.
- [36] Yuichi Goto(@yasaichi) <https://github.com/yasichi/>. Implementation of X-means clustering in Python. <https://gist.github.com/yasaichi/254a060eff56a3b3b858> 最終確認日 2018 年 1 月 16 日.
- [37] 高校数学の美しい物語. 不偏標本分散の意味と  $n-1$  で割ることの証明. <https://mathtrain.jp/huhenbunsan> 最終確認日 2018 年 1 月 17 日.
- [38] Googlemaps. <https://www.google.co.jp/maps> 最終確認日 2017 年 12 月 19 日.