

# 軌道形状クラスタリングに基づく交通量調査の自動測定方式の実装と評価

渡辺 魁<sup>†1,a)</sup> 中野 勝章<sup>†1</sup> 中沢 実<sup>†1</sup>

**概要**：近年、画像認識技術の向上により交通量調査も MOT(Multi-Object Tracking) を利用したシステムに置き換わると期待されている。MOT を利用した交通量調査では、トラッキングされた車両の軌道に対して検知線を用いて交差判定を行うことでカウントする手法が主流となっており、多くのアプリケーションで採用されている。しかし、検知線によるカウントでは予め正解の計測結果がわかっていると検知線の設定を最適化できず、ほとんどのアプリケーションでは人手による設定が求められる。そのため、設定者により計測精度にばらつきが生じたり、試行錯誤が求められる問題がある。そこで、本研究では、人手による計測結果を必要としないカウントの自動化を実現するため、車両の軌道の形状に基づいた交通量計測のためのフレームワークを提案する。さらに、提案手法の有効性を評価するため、検知線による計測結果との比較検証を実施した。

**キーワード**：Multi-Object Tracking, 交通量調査, node-moving-things-tracker, k-means++

## 1. はじめに

一般交通量調査とは、全国道路・街路交通情勢調査の一環として全国の道路の交通量及び道路現況等を調査し、道路の計画、建設、維持修繕その他の管理などについての基礎資料を得ることを目的に実施されている [1]。現状では人手による計測が主流となっている [2] がコスト削減・業務の効率化などの観点からセンサーや AI による自動計測への期待が高まっている。また、令和2年度の調査はコロナ禍の影響により延期となっており [3]、無人での交通量の計測の重要性が高まっている。既存の AI を利用したソフトウェアやサービス [4][5][6] ではいずれも車両の検出と Multi-Object Tracking(MOT) を行い、通過の有無を判定する検知線を設定し、通過台数を計測されている。

しかし、検知線によるカウントでは予め正解の計測結果がわかっていると検知線の設定を最適化することができず、ほとんどのアプリケーションでは人手による設定が求められる。特に交差点のような複雑な環境では直線道路に比べて検知線の設定する本数が増えることや、トラッキングのノイズも多くなる。そのため、設定者により計測精度

にばらつきが生じたり、設定の試行錯誤が求められる。この問題に対し、著者ら [7] は検知線の設定を自動化する手法を提案しているが、完全な自動化には未だ至っていない。

そこで本研究では、計測結果を必要としないカウントの自動化を実現するために k-means++法 [8] を用いた交通量調査の自動測定手法を提案する。提案手法では、金沢市内の2箇所の交差点を撮影した動画を node-moving-things-tracker[9] でトラッキングした結果を対象に車両のカウントを行う。また、提案手法の実用性を評価するため、先行研究の検知線によるカウントアプローチとの計測精度の比較を行なった。

本稿の構成は以下の通りである。第2章では関連研究について述べ、第3章では本研究の検証に利用したデータセットについて述べる。また、第4章では本研究で提案する軌道形状に基づくカウントアプローチについて述べる。第5章では提案手法と検知線によるアプローチとの比較検証の結果と考察を述べ、最後に6章で本研究のまとめと今後の課題について述べる。

## 2. 関連研究

### 2.1 MOT

Multi-Object Tracking(MOT) は動画内の対象とするクラスの複数の物体の追跡を行うタスクである。追跡対象はそれぞれ学習したモデルにより異なり、複数クラスを対象

<sup>1</sup> 情報処理学会

IPSJ, Chiyoda, Tokyo 101-0062, Japan

<sup>†1</sup> 現在、金沢工業大学

Presently with Kanazawa Institute of Technology

<sup>a)</sup> c6200536@st.kanazawa-it.ac.jp

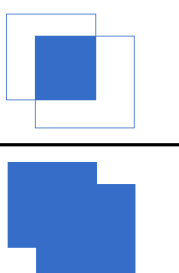
$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$


図1 IoU の計算

とする場合もある。MOT では同一のオブジェクトには各フレームで同じ ID を割り当てることでその物体の軌跡の情報を得ることができる。深層学習モデルを用いた MOT の手法は 1-shot と 2-stage に分類される。

1-shot は物体の検出と追跡を一つのモデルでエンドツーエンドで行う手法で、代表的なものとして FairMOT[10], STGT[11] などがある。2-stage は MOT を物体の検出と追跡の 2 つのタスクに分割し、それぞれのタスクに特化したモデルやアルゴリズムを利用する手法である。代表的なモデルとして物体検出部は YOLOv4[12] や Faster R-CNN[13] などの高速な物体検出モデルを用い、トラッキングは検出されたバウンディングボックス (Bbox) の情報を元に ID を割り当てる SORT[14] や DeepSORT[15], node-moving-things-tracker がある。計算資源に応じて適切なモデルを選択することでエッジ端末で利用することも可能である。

### 2.1.1 node-moving-things-tracker

node-moving-things-tracker は、リアルタイムに MOT を行うトラッキング処理の javascript 実装で V-IOU tracker[16][17] に基づいている。2.2 節で紹介する OpenDataCam では YOLOv4 による物体検出とともに利用されている。トラッキングの方法としてはフレーム間で検出された物体の Bbox の重なり具合を IoU(図 1) で算出し ID の割り当てを行い(図 2(a)), 物体の検出が途切れた際には単一のオブジェクトトラッキングモデルに切り替え(図 2(b)), 再度検出された際に IoU をもとに ID の割り当てを行う(図 2(c)) ことで物体検出モデルの偽陰性(検出漏れ)に対してロバストなトラッキング手法となっている。

## 2.2 OpenDataCam

OpenDataCam[18] はリアルタイムのビデオ解析に特化したオープンソースツールで、主に移動する物体のトラッキングやカウントのための GUI(図 3) など交通量調査に必要な様々な機能が備わっている。特に NVIDIA Jetson-Board シリーズ [19] などのエッジコンピュータでの処理に最適化されており、YOLOv4 による高速な物体検出と node-moving-things-tracker による MOT を用いて少ない

計算資源でもリアルタイムでの処理が可能となっている。

### 2.2.1 検知線によるカウントアプローチ

検知線はトラッキング結果との交差判定を行うことでそこを通過した物体の有無や車両の通過台数を計測することができる。図 3(b) では進行方向別に検知線を設定しており、適切な設定をすることでカウントを自動で行なうようになっている。OpenDataCam では検知線に対しいくつかのオプションを設定することができる。参照する過去のフレーム数を 3 としたときの OpenDataCam のカウント処理を図 4 に示す。ここでは現在のフレームから指定したフレーム間のトラッキング結果の端点を用いて検知線との線分の交差判定を行う。4 フレーム目の段階ではまだ交差していないためカウントされず、5 フレーム目になるとトラッキングの線分が検知線と交差してカウントが行われる。OpenDataCam はリアルタイムで車両のトラッキングから交差判定を行うため逐次的にカウントする仕様となっている。

## 2.3 AI を用いた交通量調査の現在

OpenDataCam の他にも、AI を利用した交通量調査のためのさまざまなサービスやツール [4][5][6] が登場しており、いずれも検知線によるカウントアプローチをとっている。実際にこれらを利用する際にはカメラの設置位置の問題や適用範囲に制約があるため、付近にカメラを設置可能な高層の建造物が無かったり、交差点の場合は対応していないケースもある。Dulbadrakh ら [20] の研究では安価な Web カメラと三脚を用いた比較的 low 所から撮影した動画を利用したシステムの提案を行なっている。中野ら [21] は交差点内に設置した複数台のカメラ映像を比較し、短時間のトラッキング結果から計測精度に影響を与える要因を評価指標としたカメラ設置位置の最適化を提案している。

著者ら [7] は先行研究において探索アルゴリズムによる検知線の設定の自動最適化手法を提案しており、人が手動で設定した場合と同等以上の計測精度を実現している。しかし、検知線の最適化のためにはあらかじめ正解のカウント数が必要であることや、検知線の設定範囲をあらかじめ指定する必要があるため、完全な自動化は実現できていない。

## 3. データセット

本研究では金沢市内の 2 箇所の交差点の動画を撮影して、そのデータを元にデータセットの作成を行なった。なお、本データの撮影は石川県ならびに金沢市からの協力の元、道路使用許可を得て実施している。

今回対象としたのは兼六園上交差点(図 5(a))と広坂交差点(図 5(b))でそれぞれ 2 度信号が切り替わる間の 5 分と 10 分の動画を利用した。撮影した動画を OpenDataCam に実装されている node-moving-things-tracker によりトラッキングを行い、今回対象とする車両のクラス(車、バイク、

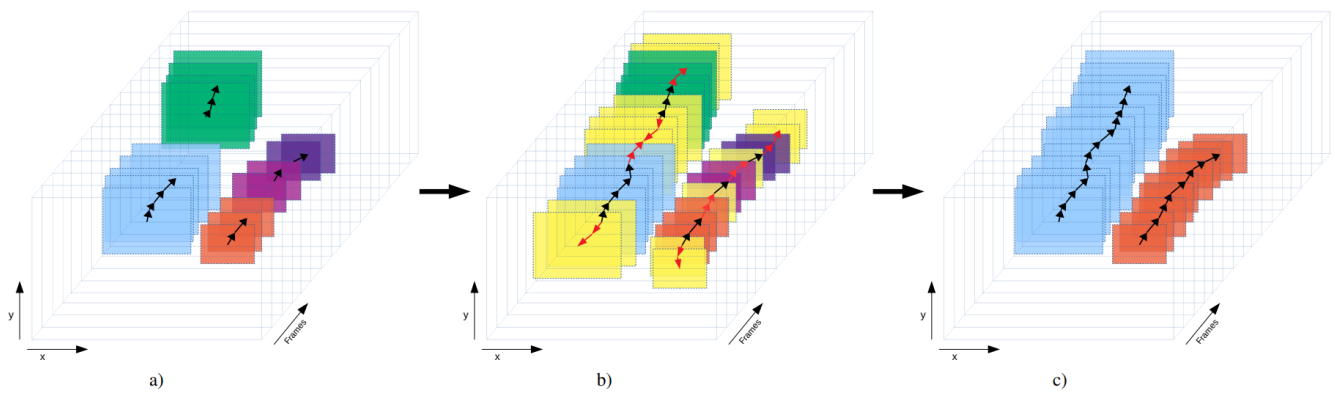
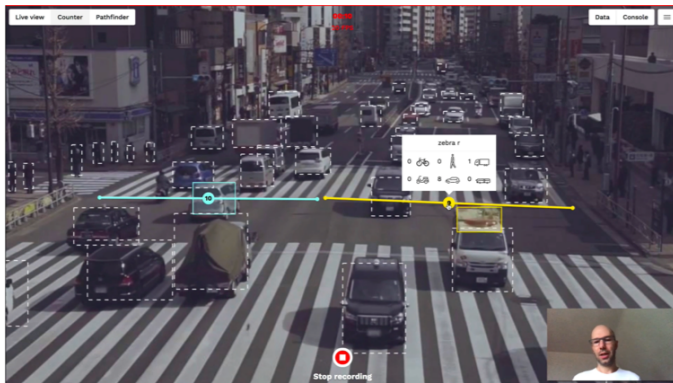


図 2 V-IOU における検出漏れ発生時の処理の流れ [17]Figure 1. より引用



(a) node-moving-things-tracker によるトラッキング結果の表示



(b) 検知線の設定画面

図 3 OpenDataCam の GUI 画面 [22]

バス、トラック) 以外のトラッキング結果を削除した。また、計測精度の評価用にそれぞれの交差点の方向別の車両の通過台数を目測して、これを正解データとした。

## 4. 提案手法

### 4.1 概要

提案手法の処理の流れ(図6)を説明する。まず、前処理として OpenDataCam によるトラッキング結果に対し不要な軌道の削除を行う。次に、各軌道の始点と終点の座標に対して k-means++法 [8] を用いたクラスタリングを実施し、それぞれの軌道がどの流入口と流出口を通過しているか

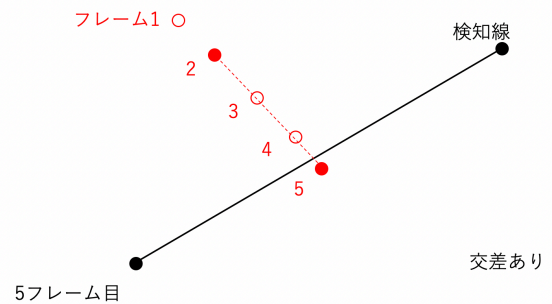
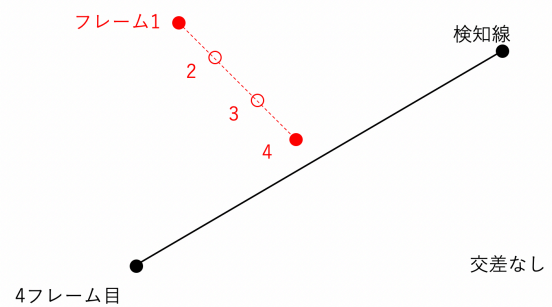


図 4 OpenDataCam のステップごとのカウント処理

調べる。その結果から方向別に軌道のカウントを行い、計測結果を得る。

### 4.2 軌道データの前処理

全てのトラッキング結果には図7で示されるような比較的短い軌道が含まれている。このようなデータは物体検出の際の誤検出やトラッキングでの id の再割り当ての失敗により発生している。これらのデータが含まれている状態では正確なクラスタリングに影響が出る恐れがあるため閾値を設定して削除する必要がある。今回は各軌道の総移動距離が 300px 未満のものを削除した。図7に削除された軌道と左上に閾値の長さを示す。なお、今回使用したビデオの解像度は 1920 × 1080px とした。





(a) 兼六園上交差点 (T 字路)



(b) 広坂交差点 (十字路)

図 5 対象となる交差点の画像



(a) 兼六園上交差点 (T 字路)



(b) 広坂交差点 (十字路)

図 7 前処理で除去された軌道とその閾値

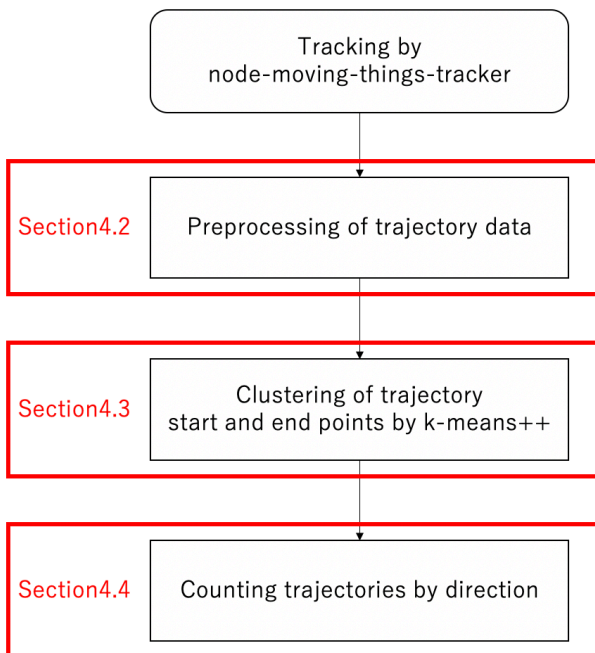
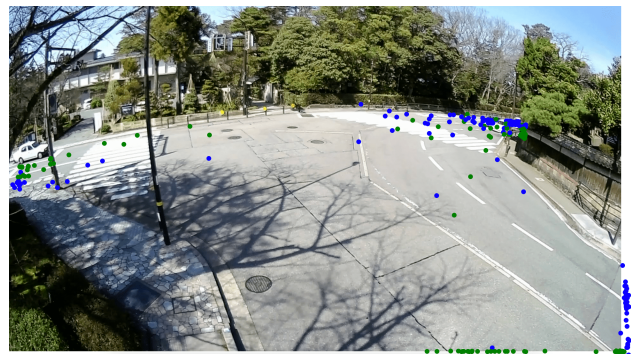


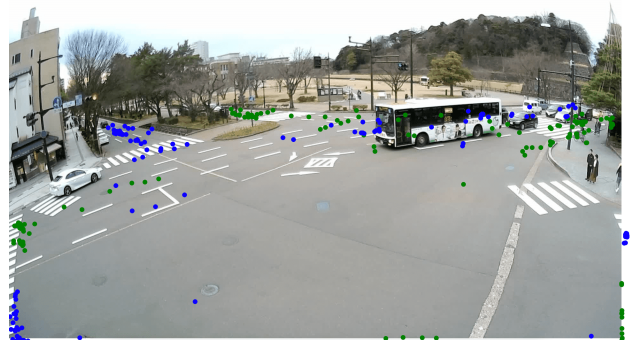
図 6 提案手法のフローチャート

### 4.3 k-means++法によるクラスタリング

k-means++法は、k-means 法の初期値依存問題の解消を目的としたアルゴリズムで、速度と精度の面で優れているとされている。k-means 法では初期値をランダムに選択しているが、k-means++法では初期のクラスターの中心はで



(a) 兼六園上交差点 (T 字路)



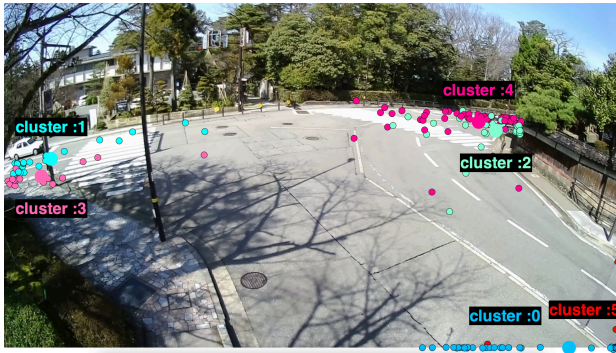
(b) 広坂交差点 (十字路)

図 8 軌道の始点と終点

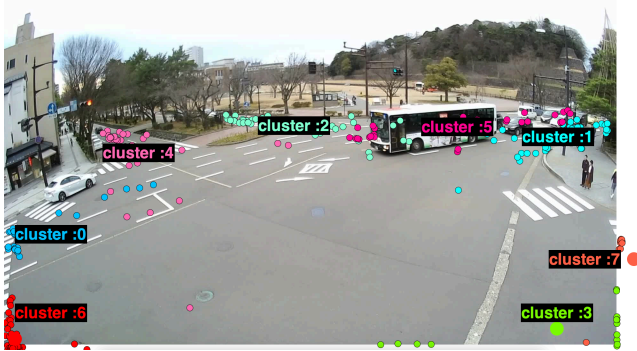
きるだけ離れていた方がよいという考えに基づき、データ点間の距離に基づいて確率的に初期値を決定する。

前処理後の軌道データに対して、図 8 の緑の点は各軌道の始点で青は終点を表している。方向別の軌道のカウント





(a) 兼六園上交差点 (T 字路)



(b) 広坂交差点 (十字路)

図9 始点と終点のクラスタリング結果

を行うために始点と終点それぞれの集合で k-means++法によりクラスタリングを行い、各軌道がどこからどこへ向かっているのか分類しカウントする。クラスタリングを行った結果を図9に示す。なお、交差点はそれぞれ三叉路と十字路であるため、クラスタ数をそれぞれ3, 4とし、始点と終点それぞれでクラスタリングを行っているため合わせて6, 8個のクラスタが得られている。

#### 4.4 軌道のカウント

図9は始点と終点のクラスタごとに色分けされた軌道を示しており、それぞれ対応する方向に向かっている軌道としてカウントされる。今回対象とした交差点のデータでは方向によっては1台も通過していない場合があったため、出入の向きを区別せずにカウントした結果で計測精度を計算する。計測精度  $acc$  を式(1)で求める。ここで、 $gt$  は目測の計測台数、 $c$  は提案手法のカウント数である。 $acc$  を方向別に求め、それらの平均値を最終的な評価指標とした。

$$acc = \frac{gt - |gt - c|}{gt} \quad (1)$$

## 5. 評価実験

### 5.1 評価方法

提案手法の実用性を評価するため検知線によるアプローチ [7] と計測精度による比較を行った。また、設定する検知線の本数は入出口に対して1本としたため、兼六園上交差点では3本、広坂交差点では4本となっている。

表1 検証結果の比較 (node-moving-things-tracker)

交差点	検知線によるアプローチ	提案手法
兼六園上 (T 字路)	0.91	0.85
広坂 (十字路)	0.81	0.95

### 5.2 実験結果

それぞれの交差点ごとの計測精度の比較結果を表1より示す。検知線によるアプローチと比べ提案手法は平均して高い計測精度を達成している。特に広坂交差点 (十字路) では提案手法が0.14高い結果となった。一方で、兼六園上交差点 (T 字路) では提案手法が劣る結果となっており、原因として id switch が多発していることが挙げられる。

id switch とは、複数の追跡物体が交差もしくは重なった際に同一物体を識別する id が入れ変わるトラッキングのミスのことである。

### 5.3 計測結果に対する id switch の影響

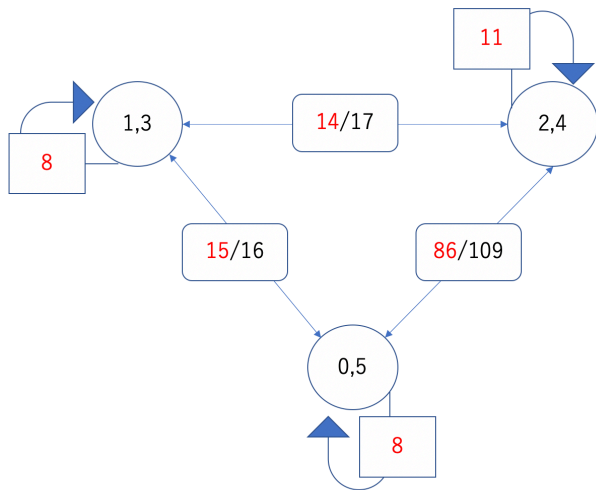
提案手法の方向別のカウントの詳細を図10に示す。各ノードは交差点の入出口ごとにクラスタの id をまとめたもので、ノード間にある数値は分子が提案手法によるカウント数で赤色で表記し、分母は目測で計測した際のカウント数となっている。また、同じノードに繋がっているエッジ (自己ループ) は同じ入出口となっている軌道の数を示しており、最終的なカウント数には含まれない。兼六園上交差点では広坂交差点に比べて多くの自己ループが存在しており、本来カウントされるべき軌道が含まれていると考察できる。

兼六園上交差点において自己ループに分類された軌道を図11に示す。これらの軌道は大きく2つに分類でき、1つ目は比較的短い軌道で図11の (b),(c) で確認できる。このような軌道が生成される要因としては検出ミスやトラッキングが失敗などが挙げられる。2つ目はUターンを描いている軌道で、特に図11(a)のほとんどはこの軌道に分類でき、クラスタ0,5からクラスタ2,4間のカウント数が減少している大きな要因となっている。このような軌道が生成されるのは、実際に車両がUターンしているのではなく、交差点の入出口でトラッキングの id switch が発生し2つの車両に同一の id が割り当てられたためである。

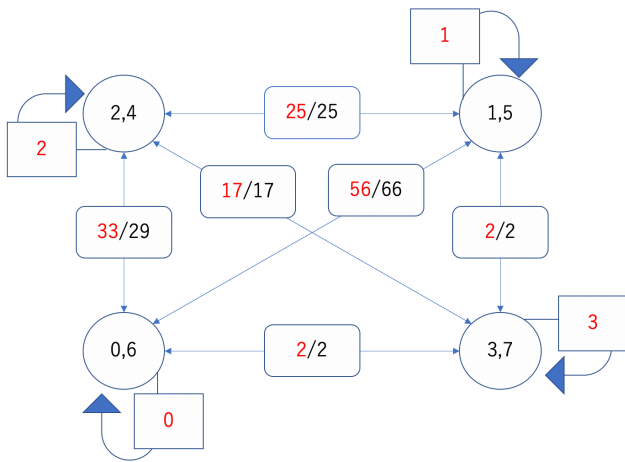
id switch の問題は同じ入出口だけに留まらず、本来のルートとは違うルートを経由した際に間違ったカウントがされてしまう恐れもあるため、正確なカウントを行う際に重要な課題となる。

## 6. まとめ

本研究では、検知線を用いたアプローチのように事前に交通量がわかっても正確なカウントを実現するために、軌道の始点と終点に対して k-means++法によりクラスタリングし、方向別の軌道のカウントする手法を提案



(a) 兼六園上交差点 (T 字路)



(b) 広坂交差点 (十字路)

図 10 方向別のカウント数



(a) クラスタ 0 から 5 への軌道



(b) クラスタ 1 から 3 への軌道



(c) クラスタ 2 から 4 への軌道

図 11 自己ループに分類された軌道

し、検知線によるアプローチとの精度の比較を行なった。結果として提案手法は検知線によるアプローチに比べて平均して高い計測精度を得ることを確認した。特に十字路のような比較的複雑なシーンでは検知線によるアプローチを大きく上回る精度が期待できることがわかった。しかし、id switch が発生している軌道に対しては検知線に比べてカウントの減少傾向が見られ、平均計測精度が悪化する問題が見られる。

今後の予定としては、計測精度を向上させるとともにロバスト性の高い計測を実現するため、id switch の発生している軌道を検出し、元の複数台の車両の軌道に復元することを検討する。また、現状では 2 箇所の交差点でしか検証できていないため、他研究で用いられているデータセットの利用や計測精度の比較を実施する。さらに、前処理の段階で除外している短い軌道を復元することで検知線や提案手法では捉えきれない軌道のカウントを実現したい。

### 参考文献

- [1] 国土交通省：一般交通量調査について，入手先 (<http://www.mlit.go.jp/road/census/h22-1/data/kasyorep.pdf>) (参照 2021-11-18).
- [2] 国土交通省：平成 27 年度 全国道路・街路交通情勢調査 一般交通量調査結果の概要について，入手先 (<https://www.mlit.go.jp/common/001187536.pdf>) (参照 2021-11-18).
- [3] 国土交通省：令和 2 年度全国道路・街路交通情勢調査の延期について，入手先 (<http://www1.mlit.go.jp/road/ir/ir-council/ict/pdf04/01.pdf>) (参照 2021-11-18).
- [4] Scorer Traffic Counter，入手先 (<https://www.scorer.jp/products/scorer-traffic-counter/>) (参照 2021.11.24).
- [5] 映像解析サービス&ソフトウェア ビューリーダー，入手先 (<https://www.kotsu-chosa.com/vr-site/vr-top.html>) (参照 2021.11.24).
- [6] 簡単にらくらく通行量分析，入手先 (<https://www.chess-inc.com/odpt/>) (参照 2021.11.24).
- [7] 渡辺 魁，中野 勝章，中沢 実：MOT を用いた交差点における交通量調査のカウンターラインの最適化手法の提案，情報処理学会研究報告，2021-DPS-189，Vol.2021，No.14，pp.1-6 (2021).



- [8] Vassilvitskii, S., Arthur, D.: k-means++: The advantages of careful seeding, In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms ,pp. 1027-1035(2006).
- [9] node-moving-things-tracker (Github), <https://github.com/opendatacam/node-moving-things-tracker>, (参照 2022.7.19).
- [10] Zhang, Yifu, et al: Fairmot: On the fairness of detection and re-identification in multiple object tracking, International Journal of Computer Vision 129.11 (2021)
- [11] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, Zicheng Liu: TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking, <https://arxiv.org/abs/2104.00194v2> (参照 2022.7.19).
- [12] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao: YOLOv4: Optimal Speed and Accuracy of Object Detection, <https://arxiv.org/abs/2004.10934> (参照 2022.7.19).
- [13] Ren, Shaoqing, et al.:Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).
- [14] Bewley, Alex, et al.: Simple online and realtime tracking, 2016 IEEE international conference on image processing (ICIP), IEEE, (2016).
- [15] Wojke, Nicolai, Alex Bewley, and Dietrich Paulus.: Simple online and realtime tracking with a deep association metric, 2017 IEEE international conference on image processing (ICIP), IEEE, (2017).
- [16] Erik Bochinski Volker, Eiselein, Thomas Sikora: High-Speed Tracking-by-Detection Without Using Image Information, <http://elvera.nue.tu-berlin.de/files/1517Bochinski2017.pdf>, (参照 2022.7.19).
- [17] Erik Bochinski, Tobias Senst, Thomas Sikora: Extending IOU Based Multi-Object Tracking by Visual Information <http://elvera.nue.tu-berlin.de/files/1547Bochinski2018.pdf>, (参照 2022.7.19).
- [18] OpenDataCam 3.0.2 – An open source tool to quantify the world (Github), <https://github.com/opendatacam/opendatacam>, (参照 2022.7.19).
- [19] Jetson Nano 開発者キット, 入手先 (<https://www.nvidia.com/ja-jp/autonomous-machines/embedded-systems/jetson-nano/education-projects/>)(参照 2022.7.19).
- [20] Temuulen Dulbadrakh, 鈴木雅人, 北越大輔, 西村 亮: YOLO 及び MOT を用いた車種別交通量調査の自動化のための研究, 第 83 回全国大会, 7Q-07, 情報処理学会, 2021.3 (参照 2022.7.19).
- [21] 中野勝章, 渡辺魁, 中沢実: 交差点の交通量調査におけるカメラ設置最適化の一提案, 研究報告モバイルコンピューティングと新社会システム (MBL) 2022.15 (2022): 1-6.
- [22] OpenDataCam 2.0 – User Interface Walkthrough, 入手先 (<https://vimeo.com/346340651>)