

# MOT活用による方向別交通量調査における カウンターライン最適化手法の構築と評価

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

受付日 2022年4月13日, 採録日 2022年11月8日

**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, トラッキングした対象の軌跡に対して交差判定により通過台数を計測するカウンターラインを設定する手法が主流となっている. しかし, 現状ではカウンターラインの設定を補助する機能が不足しており, 最適な設定にするためには人手による試行錯誤が求められる. そのため, 計測までにかかるコストの増加や設定者によって計測精度にばらつきが出る恐れがある. 特に交差点における方向別の計測などの複雑なシーンでは複数本のカウンターラインを同時に最適化する必要があるため, より困難なタスクとなる. そこで, 本研究ではカウンターラインの設定作業の一部を探索アルゴリズムにより自動化することで, 設定者に対する依存と負担を削減し, 対象ツールの実用性を向上させる. また, 実用性の評価のため, 金沢市内の 2 カ所の交差点を撮影した動画に対して node-moving-things-tracker と DeepSORT の 2 つのトラッキング手法でのトラッキングを行い, 人手による設定と提案手法による設定の比較検証を実施した.

**キーワード:** Multi-Object Tracking, 交通量調査, OpenDataCam, node-moving-things-tracker, DeepSORT

## Development and it's Evaluation of a Counterline Optimization Method for Directional Traffic Surveys Using MOT

KAI WATANABE<sup>1</sup> KATUAKI NAKANO<sup>1</sup> MINORU NAKAZAWA<sup>1,a)</sup> KANAME NAGANUMA<sup>1</sup>

Received: April 13, 2022, Accepted: November 8, 2022

**Abstract:** In recent years, with the improvement of image recognition technology, traffic surveys are expected to be accurate by systems using MOT (Multi-Object Tracking). The most common method used in MOT-based traffic surveys is to set a counter line to measure the number of vehicles passing through a tracked object by determining the number of intersections. However, at present, there is no function to support the setting of the counter line, and repeated trial and error is required to make the optimum setting. This increases the cost of measurement and may lead to variations in measurement accuracy depending on the person who sets up the system. This task is complicated in complex scenes such as intersections, where multiple counter lines must be optimized simultaneously. Therefore, in this research, by automating a part of the counter line setting work by the search algorithm, the dependence and burden on the setter are reduced, and the practicality of the target tool is improved. To evaluate the practicality of the proposed method, two tracking methods, node-moving-things-tracker, and DeepSORT, were used to track videos of two intersections in Kanazawa City, and a comparison was made between the manual settings and the settings using the proposed method.

**Keywords:** multi-object tracking, traffic surveys, opendatacam, node-moving-things-tracker, deepSORT

<sup>1</sup> 金沢工業大学  
Kanazawa Institute of Technology, Nonoichi, Ishikawa 921-8501, Japan

<sup>a)</sup> nakazawa@infor.kanazawa-it.ac.jp

### 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および道路現況などを調査し、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。既存の交通量調査のための AI を利用したソフトウェアやサービス [7], [8], [9] 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カウンターラインを設定し、通過台数の計測をするのが主流となっている。カウンターラインによる計測ではトラッキングの精度と適切なカウンターラインの設定が必要であり、どちらが欠けても最終的な計測精度に影響が出るため重要な問題となる。そのため、カウンターラインの設定者は比較的単時間の動画に対して目測により通過車両の計測を行い、それを正解データとしてカウンターラインを調整した後、長時間計測を行うといった方法がとられている。

しかし、現状ではカウンターラインの設定を補助する機能が不足しており、設定者が経験と勘に基づき設定を行うほかないため、最適な設定にするためには人手による試行錯誤が求められる。一般的にはトラッキングのミスが発生するため、カウンターラインのカウント数と実際の通過台数が一致することは稀であり、設定者はカウンターラインの設定が最適なのか判断できない。そのため、より多くの試行回数を必要とし、計測までにかかるコストの増加や設定者によって計測精度にばらつきが出る恐れがある。特に交差点などの複雑なシーンでは複数本のカウンターラインを同時に最適化する必要があるため、より困難なタスクとなる。

本研究では、本計測前の短時間計測におけるカウンターラインの調整にかかるコストを削減するとともに設定者の経験に依存しないことを目的とし、ランダムサーチを用いたカウンターラインの設定の最適化手法を提案する。提案手法では、金沢市内の 2 カ所の交差点を撮影した動画をトラッキングした結果を対象にカウンターラインの最適化を行う。提案手法の汎用性を示すため node-moving-things-tracker と DeepSORT の 2 つのトラッキング手法を用いて検証を行った。また、提案手法の実用性を評価するため、人手でのカウンターラインの設定との比較を行った。

本論文の構成は以下のとおりである。2 章では関連研究について述べ、3 章では本研究の検証のために作成したデータセットについて述べる。また、4 章では本研究で提案するカウンターラインの最適化手法について説明する。5 章では提案手法と人手による設定との比較検証の結果について述べ、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 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 1 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] 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の情報をもとに ID を割り当てる SORT [13] や今回利用した node-moving-things-tracker [5], DeepSORT [14] 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できる。

#### 2.1.1 node-moving-things-tracker

node-moving-things-tracker は、リアルタイムに MOT を行うためのトラッキング処理の JavaScript 実装で V-IOU tracker [17], [18] 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 で算出し ID の割当てを行い (図 1 (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替え (図 1 (b)), 再度検出された際に IoU をもとに ID の割当てを行う (図 1 (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モデルとなっている。

#### 2.1.2 DeepSORT

SORT は物体検出モデルにより検出された対象が次のフレームでどこに位置するかをカルマンフィルタにより予測を行い、新たに検出結果が得られた際にそのバウンディングボックスとの IoU をもとに ID の割り振りを行うトラッキングモデルである。これは高速に動作する一方、バウンディングボックスの位置情報のみを利用しているため、追跡対象の物体が障害物の影に隠れてしまい検出が途切れてしまう occlusion が発生したり別の追跡対象の物体と交差する際に ID の割当てが正常に行われなかった問題がある。DeepSORT は SORT を改良した手法で、SORT では検出された物体の位置情報のみを利用していたのに対し、DeepSORT では新たに深層学習モデルを使用し検出された物体の特徴ベクトルと追跡対象の特徴ベクトルを取得し、それらのコサイン類似度を計算することで視覚的な

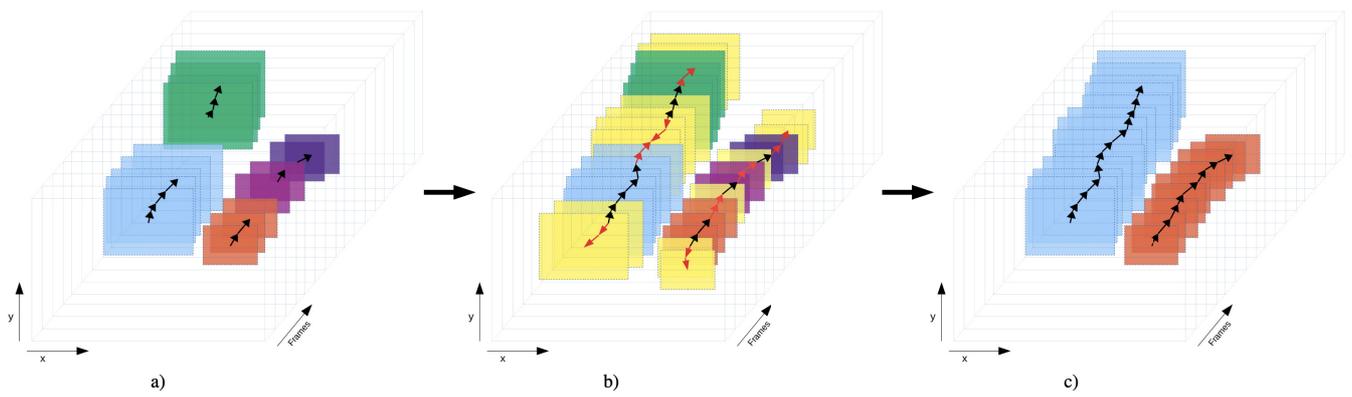


図 1 V-IUO における検出漏れ発生時の処理の流れ. 文献 [18] Figure 1. より引用  
 Fig. 1 Process flow in case of detection leakage in V-IUO. Ref. [18] Figure 1.



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



(b) カウンターラインの設定画面

図 2 OpenDataCam の GUI 画面 [21]

Fig. 2 GUI screen of OpenDataCam [21].

特徴量を考慮した ID の割り振りが可能になる. そのため occlusion や物体の交差による id の switching に対してもよりロバストなモデルとなっている.

## 2.2 OpenDataCam とカウンターライン

### 2.2.1 OpenDataCam

OpenDataCam 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 ツール (図 2) などさまざまな機能が備わっている. 特に NVIDIA JetsonBoard シリーズ [16] 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. 図 2(a) は node-moving-things-tracker によるトラッキング結果を表示したもので, 同一物体ごとに異なる色で軌道が描画されている.

### 2.2.2 カウンターライン

カウンターライン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 2(b) では進行方向別にカウンターラインを設定しており, トラッキング結果と交差するように設定することでカウントを自動で行うようになっている. OpenDataCam ではカウンターライン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 3 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, カウンターライン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カウンターラインと交差し, 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 MOT と交通量調査

MOT 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が登場している [7], [8], [9]. 実際にこれらを利用する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なかったり, 交差点などの場合は対応していないケースもある.

Dulbadrakh ら [15] の研究では安価な Web カメラと三脚を用いた比較的低位から撮影した動画を利用したシステムの提案を行っている. Andres ら [20] は, あらかじめラベル付けされた関心領域内を通過している車両の軌道の始点

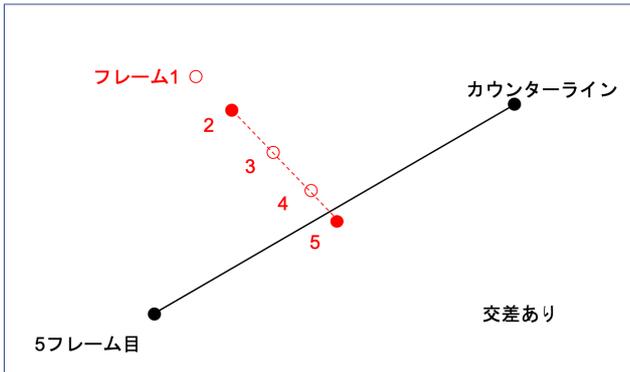
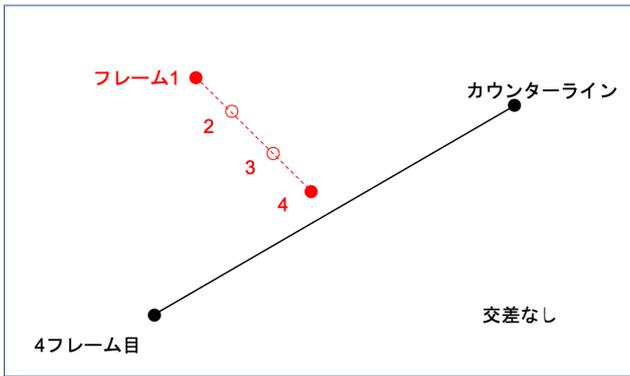


図 3 OpenDataCam のステップごとのカウント処理

Fig. 3 Step-by-step counting process for OpenDataCam.

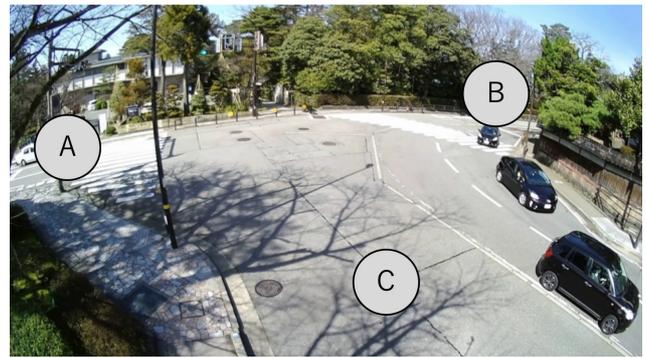
と終点のベクトルに対してカウンターラインを設定し、方向別の車両のカウントを行っている。

いずれの手法もカウンターラインの設定は人手で行うことが前提であり、異なる交差点環境に適用する際にはアンテーションが必要となる。正確なカウントを行うには適切にカウンターラインが設定されている必要があり、設定者が経験や勘で設定するか、短期間の計測結果をもとにカウンターラインを調整するといった方法がとられる。しかし、一般的にはトラッキングのミスが発生するため、カウンターラインのカウント数と実際の通過台数が一致することは稀であり、設定者はカウンターラインの設定が最適なのか判断できない。そのため、より多くの試行回数を必要とし、計測までにかかるコストの増加や設定者によって計測精度にばらつきが出る恐れがある。特に実地調査では調整できる時間が限られているため、検証は困難である。

本研究では、短期計測結果に基づきカウンターラインの調整を自動化することで、人手による試行錯誤にかかるコストを削減し、現実的な時間で最適な設定を探索する。

### 3. データセット

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



(a) 兼六園上交差点 (T 字路)



(b) 広坂交差点 (十字路)

図 4 対象となる交差点の画像

Fig. 4 Images of targeted intersections.



図 5 トラッキング結果

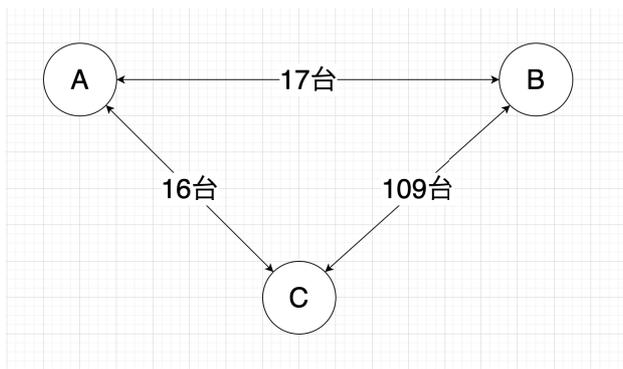
Fig. 5 Tracking results.

#### 3.1 対象となる交差点のトラッキングデータ

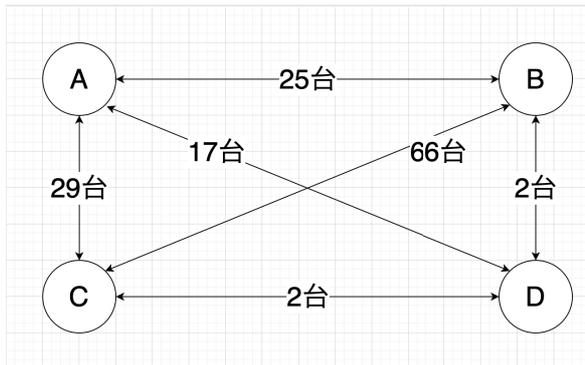
今回対象としたのは兼六園上交差点 (図 4(a)) と広坂交差点 (図 4(b)) で 2020 年 3 月 17 日の午前 9:00 から撮影されたもので、それぞれ 2 回信号が切り替わるまでの動画を利用した。動画を node-moving-things-tracker と DeepSORT によりトラッキングし、今回対象とする車両のクラス (車、バイク、バス、トラック) 以外のトラッキング結果を削除した。その結果を図 5 に示す。

#### 3.2 正解データの作成

カウンターライン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、これを正解データとした。計測は著者が 1 人で行い、カウント漏れがないか撮影した動



(a) 兼六園上交差点の車両の通過台数



(b) 広坂交差点の車両の通過台数

図 6 各交差点の交通量の集計結果

Fig. 6 Total traffic volume at each intersection.

画を繰り返し確認を行った。それぞれの交差点での計測結果を図 6 に示す。

## 4. 提案手法

### 4.1 提案手法の概要

提案手法の処理の流れ(図 7)を説明する。まず、最初に前処理として、OpenDataCam によるトラッキング結果に対して不要なトラッキングの削除を行う。次に、人手により対象とする交差点の侵入位置の情報を与え、トラッキング結果のクラスタリングを行う。その後、侵入位置ごとにカウンターラインの設定を探索し、正解値に最も近くなるような組合せを見つける。

### 4.2 不要なトラッキング結果の削除

トラッキング結果には図 8 で示されるような比較的短いフレーム数でのみ検出されるデータが含まれている。これらのデータは物体検出の際の誤検出やトラッキングでの id の再割当ての失敗により発生する。このような異常なトラッキング結果は複数のカウンターラインではほとんど検出されないため、削除しても最終的な方向別のカウントには影響が出ない。一方でカウンターラインの候補を選定する際にはこれらのデータの存在が問題となる。正常なトラッキング結果をカウントできていないにもかかわらず、異常なトラッキング結果を含んだ見かけ上のカウント数が

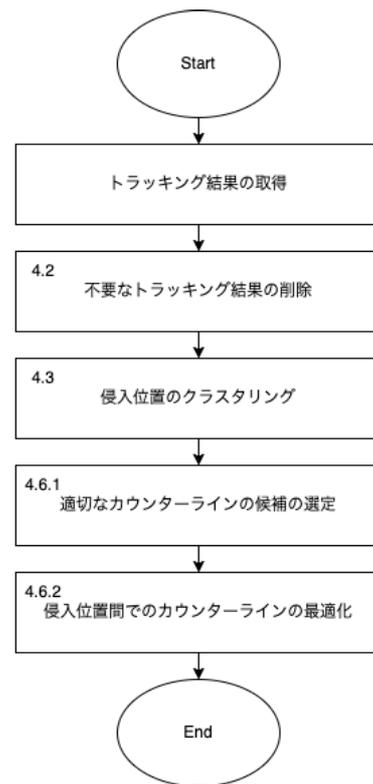


図 7 提案手法のフローチャート

Fig. 7 Flowchart of the proposed method.



図 8 不要なトラッキング結果

Fig. 8 Unnecessary tracking results.

多いカウンターラインが採用され、最終的な計測精度が悪化する恐れがあるため事前に削除しておく必要がある。今回は閾値を 100 フレームとし、検出フレーム数がそれ以下の id はトラッキング結果から削除した。閾値の設定は実際に削除されたトラッキング結果を確認し、正常なトラッキング結果を削除しない範囲で行った。なお、今回利用した動画は 30 フレーム/s となっている。

### 4.3 侵入位置のクラスタリング

カウンターラインの設置数と設置位置を指定するため人手によりトラッキング結果のクラスタリングを行う。これは後にカウンターラインの評価を行う際に、設置したカウンターラインと対象としているクラスタを紐づける必要が

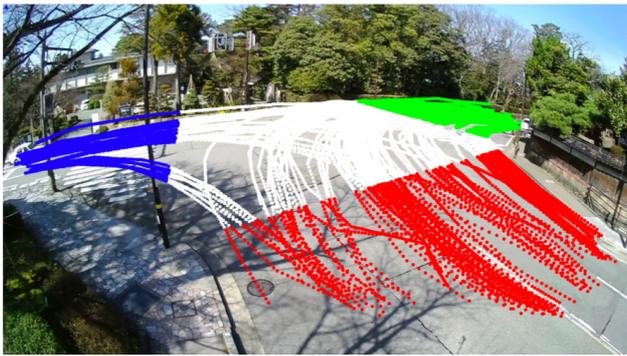


図 9 トラッキング結果のクラスタリング  
Fig. 9 Clustering of tracking results.

あるため必要となる。クラスタリングは直線で交差点の中心部と侵入位置を分割し、対応するカウンターラインが識別できるようにクラスタ id を割り振った。クラスタ id ごとに色分けされたトラッキング結果を図 9 に示す。

#### 4.4 探索するカウンターラインのパラメータ

提案手法では、カウンターラインは線分ではなく直線で設定する。直線の場合、探索するパラメータは中心座標と傾きの 2 つであるのに対し、線分の場合、直交座標ではカウンターラインの端点を指定する必要があり、その範囲の指定が難しく、極座標系では探索するパラメータ数が中心座標、傾き、線分の長さの 3 つになるが、線分の長さを変更しても直線の場合と比較してカウントできる領域が減少するだけで計測上のメリットはないため、パラメータ数の少ない直線でのカウント方式をとる。探索範囲として中心座標はクラスタリングされたエリアの点の中から選出を行い、直線の傾きは  $0^\circ \sim 180^\circ$  とした。

#### 4.5 カウントの一括処理

OpenDataCam でのカウント処理は 2.2.2 項で説明したように逐次的に行うため、1 回あたりのカウントにかかる時間が長くなるので何度もカウンターラインを引き直す際は非効率的である。そこで、提案手法ではトラッキング結果が得られた後、一括でのカウント処理を実装した。

カウントの一括処理について図 10 を用いて説明する。まず、設定したカウンターラインをもとに 2 つの領域 (A, B) に分割する。次に、それぞれの領域に含まれる共通のトラッキング結果の  $id(A \cap B)$  を求める。カウンターラインを通過していればその id はどちらの領域にも含まれているはずなので、共通の id の個数をカウント数とする。

#### 4.6 パラメータの最適化

各入出口におけるカウンターラインのカウント数が目測のカウント数と一致するような設定をランダムサーチにより探索すると図 11 の右側にある 2 本の線のように対象ではない領域でカウントを行うような引き方をしてしまう恐

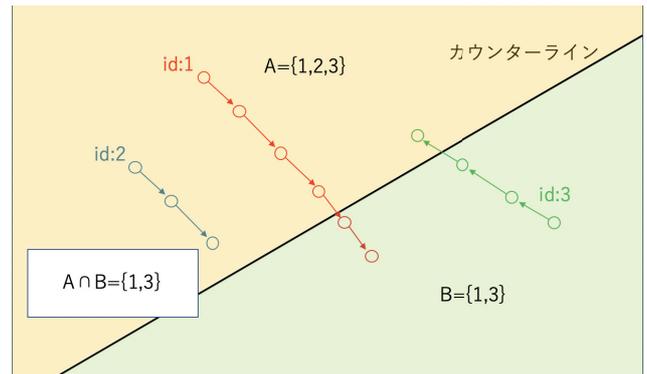


図 10 カウントの一括処理  
Fig. 10 Batch processing of counts.

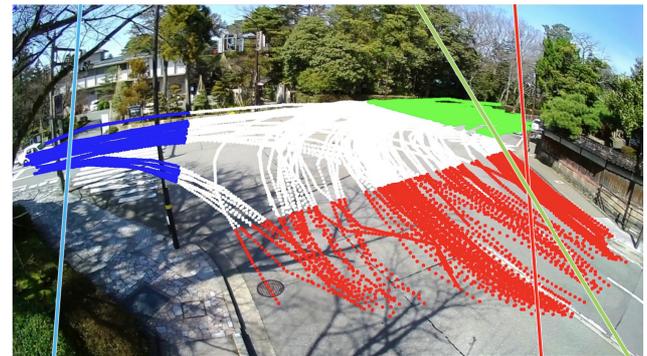


図 11 不適切なカウンターラインの例  
Fig. 11 Examples of improper counter lines.

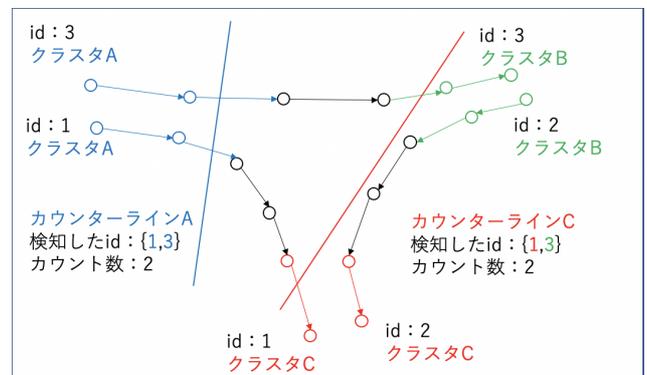


図 12 対象外の領域をカウントするカウンターラインの例  
Fig. 12 Example of a counter line counting out-of-scope areas.

れがある。

このような対象外の領域をカウントするカウンターラインが不適切な理由を図 12 を用いて説明する。図 12 ではクラスタ A を対象とするカウンターライン A とクラスタ C を対象とするカウンターライン C が設定されており、カウンターライン C は対象外のクラスタ B の軌道までカウントしている。このとき各クラスタに属する id は 2 つで、各カウンターラインのカウントも 2 つと一致している。しかし、カウンターラインで検知された共通の id が 1 と 3 であるため、クラスタ A から C の間を通過している車両は 2 台と判定されてしまい、実際に通過している台数とは一

致しない. このように, 各入出口を通過している台数と一致するようなカウンターラインの引き方はいくらでもできてしまうため, 対象外の領域をカウントしてしまうと方向別のカウントをする際に正常なカウントが行われないため不適切である.

そのため提案手法では最適なカウンターラインのパラメータの探索を2段階で行う. 1段階目では不適切なカウンターラインを除外し, 適切なカウンターラインの候補の選定を行う. 2段階目では候補の中から最適なカウンターラインの組合せを探索する.

#### 4.6.1 適切なカウンターラインの候補の選定

4.3節で行ったクラスタリング結果を用い領域ごとにカウンターラインの探索を行う. 今回は探索アルゴリズムのランダムサーチを用いて探索を実施した. ランダムサーチはローカルミニマムに陥らずに並列性を持って検索が可能な仕組みであるため, 探索パラメータ数とパラメータがとりうる範囲が限定的な場合は, 有用な方法として利用可能である. 他のブラックボックス最適化手法には遺伝的アルゴリズムやベイズ最適化などがあるが並行性が確保できないことや毎回フィードバックを必要とするため時間を要する. 今回の問題では探索するパラメータ数が2個と少なく, カウントの試行時間が非常に短いためランダムサーチを採用した.

適切なカウンターラインの候補として対象外の領域をカウントせず, 対象となる領域でのカウント数が最大となるようなカウンターラインの設定を探索. これはカウント漏れが計測精度の悪化の直接的な原因になるためである. このようなカウンターラインの設定を探索するため, カウンターラインを式(1)で評価する. ここで,  $C_T$  は対象のエリアでのカウント数で  $C_O$  は対象外のエリアでのカウント数である. 対象の領域でのカウント漏れがなく, 対象外の領域をカウントしないようなカウンターラインの評価指標  $E_1$  が最も高い値をとるため, この値が大きいものから候補として採用する. 図13に評価指標によるカウンターラインの評価の例を示す.

$$E_1 = C_T - C_O \quad (1)$$

ランダムサーチの結果, 1,000回ほど探索を行った時点で評価指標の値が頭打ちとなる設定が複数パターン得られた. 探索結果にはランダム性がともなうため, 探索回数を多く見積もることで同様の結果が得られるよう, 今回は各領域ごとに5万回探索を行った. また, 各領域において評価指標の高い10パターンを候補として保持した.

#### 4.6.2 侵入位置間でのカウンターラインの最適化

4.6.1項で探索した各領域でのカウンターラインの候補から最適なカウンターラインの組合せを総当たりで探索する. 各領域の候補から1つずつカウンターラインを選択し, それぞれのカウンターライン間での通過台数をもとに

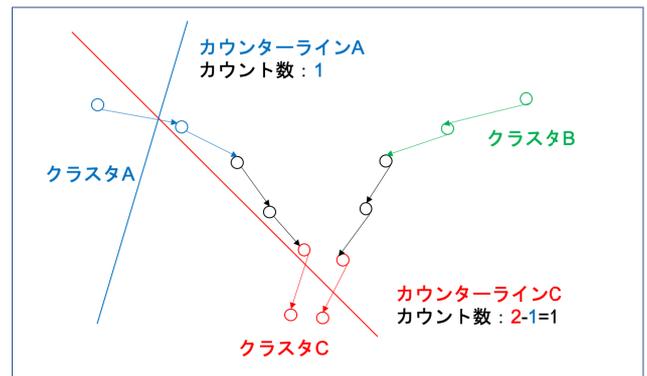


図13 カウンターラインのカウント数の計算  
Fig. 13 Calculation of counter line counts.

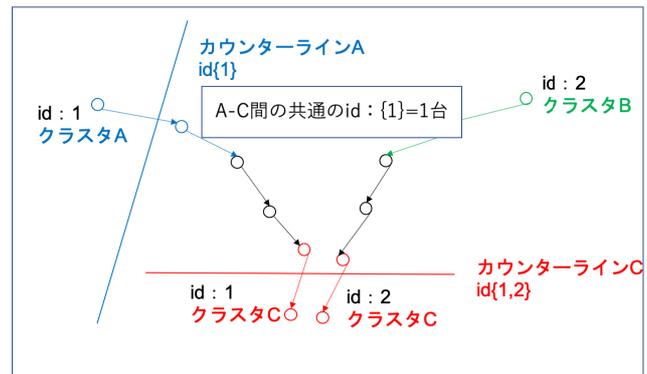


図14 カウンターライン間の通過台数の集計方法  
Fig. 14 Method of totaling the number of units passed between counter lines.

カウンターラインの評価を行う. カウンターライン間の通過台数は図14に示されるように, それぞれのカウンターラインでカウントされた共通のidの数とする.

車両の計測結果の一般な評価指標として MAPE [19] が使われているため, 評価指標  $E_2$  を以下の式(2)より求める. ここで,  $GT$  は目測した車両の台数で  $ES$  はカウンターラインによりカウントした推定値である.

$$E_2 = 1 - MAPE = 1 - \frac{|GT - ES|}{GT} \quad (2)$$

それぞれのカウンターライン間の計測精度の平均値を計算し, 候補の中で平均計測精度が最も高いものを最終的なカウンターラインとし, 探索を終了する.

## 5. 評価実験

### 5.1 探索と評価に用いるデータセットについて

カウンターラインによる計測は目測による長期計測のコスト削減のために利用されるが, カウンターラインの評価には実際の計測結果が必要である. そのため, 人手でカウンターラインを設定する際にも比較的短期間における計測を目測で行い, その結果をもとにカウンターラインを調整し, 長期計測を行う方法がとられる. 本研究の目的はこの短期間におけるカウンターラインの調整の自動化であるた

め、カウンターラインの探索と評価には同じデータセットを用いる。

## 5.2 評価方法

提案手法の汎用性の検証のため、node-moving-things-tracker と DeepSORT の 2 つのトラッキング手法に対してカウンターラインの設定を行う。node-moving-things-tracker は OpenDataCam に実装されているものを利用した。DeepSORT はインターネット上で公開されている学習済みモデルを使用した [6]。物体検出モデルは YOLOv5x を使用し、トラッキングモデルのハイパーパラメータとして max age と  $\lambda$  を調整した。max age は物体の追跡が途切れてから何フレームの間 id を保持するかを指定するもので、この値を超えた時点でシーンから離れたと見なし、割り振られる id の集合から削除される。大きすぎる値になると画面外に消えた物体の id が保持され新たに検出された物体に同じ id が割り振られる可能性があるため今回は 30 フレームとした。  $\lambda$  は式 (3) で示されるコスト関数のハイパーパラメータで、この関数は検出された物体に ID を割り振る際に使用される。  $d1$  はカルマンフィルタによる予測と検出物体のマハラノビス距離を 2 乗したもので、追跡対象の動きの情報を統合するために使用されている。  $d2$  は追跡対象と検出物体の特徴ベクトルのコサイン類似度になっており、  $\lambda$  はどちらの影響を受けるか決定するためのハイパーパラメータである。

$$\text{コスト関数} = \lambda * d1 - (1 - \lambda) * d2 \quad (3)$$

先行研究ではカメラに動きのあるタスクでは  $\lambda$  は 0 が妥当とされていたが本研究ではカメラ位置は固定してトラッキングを行うため、トラッキングミスが減るように  $\lambda$  を調整し、最終的に値を 0.2 に設定した。

今回の検証では以下の環境でプログラムを動作させて実行時間を計測した。

- PC: MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports)
- OS: macOS Big Sur 11.6.5
- CPU: 2.7 GHz クアッドコア Intel Core i7
- メモリ: 16 GB 2133 MHz LPDDR3

提案手法の有効性を示すために 4.3 節にあるトラッキング結果のクラスタリングに用いた境界線をカウンターラインに決定した際の計測結果との比較を行う。また、実用性を評価するため人手によるカウンターラインの設定との計測精度の比較を行った。同データセットを対象に被験者として交差点での実地経験の経験がある教員 1 人と経験のない学生 4 人にそれぞれ 5 回ずつカウンターラインの設定を行ってもらい、その中で最も高かったものを採用した。今回の実験では表 2 と表 3 における人手による最も高い精度を達成したのは学生によるもので、経験者であっても最

表 1 提案手法の実行時間

Table 1 Execution time of the proposed method.

交差点	実行時間
兼六園上 (T 字路)	40.4 s
広坂 (十字路)	299.6 s

表 2 検証結果の比較 (node-moving-things-tracker)

Table 2 Comparison of verification results (node-moving-things-tracker).

交差点	人手 (平均)	人手 (最高)	提案手法	境界線
兼六園上	84.5%	91%	91%	87.3%
広坂	79.2%	80%	81%	81%

表 3 検証結果の比較 (DeepSORT)

Table 3 Comparison of verification results (DeepSORT).

交差点	人手 (平均)	人手 (最高)	提案手法	境界線
兼六園上	86%	88%	92%	87.6%
広坂	73.2%	79%	79%	79%

適なカウンターラインの設定は容易ではないといえる。

## 5.3 実験結果

まず、提案手法が各交差点における最適なカウンターラインの探索にかかった時間を表 1 に示す。2 つの交差点で探索時間に大きな差が出ているのは、設定するカウンターラインの本数が増えた影響でカウンターラインの組合せのパターン数が 10 の 3 乗から 10 の 4 乗に増えたためである。人手によるカウンターラインの設定では 5 パターンのカウンターラインを設定するのにも 3 から 5 分程度の時間がかかっており、実証実験の現場では限られた時間で最適な設定にするため何度も引き直すのは困難である。提案手法はカウンターラインの設定の試行錯誤にかかる工程を自動化するとともに探索にかかる時間を大幅に削減することができ、現実的な時間で実行可能だということが分かった。

次に、トラッキング手法別でのそれぞれの交差点ごとのカウンターラインの計測精度の比較結果を表 2 と表 3 で示す。トラッキング手法を変更した際の計測精度については大差がないことが分かる。また、交差点ごとの人手による設定と提案手法による設定を比較するといずれにおいても提案手法が人手による設定と同等、もしくはそれを上回る結果となった。また、境界線をカウンターラインに設定したところ、兼六園上交差点においては提案手法と比較して精度の悪化が見られた。これは設定した境界線の付近に他の領域に属する軌道が含まれていた為で、境界線を大きくするとりすぎている場合でも提案手法はより適切なカウンターラインを設定できることが分かった。

本研究では交差点の出入口に 1 本ずつカウンターラインを設定するという比較的簡単な問題設定でカウンターラインの最適化を行った。兼六園上交差点のような比較的単純

な交差点では、カウンターラインを用いることで90%以上の精度を出すことが確認できたため有用といえる。一方で、広坂交差点では80%前後となっており、実用化に向けては課題が残っている。計測精度が悪化した要因としては、方向別でのカウントを行う際にカウンターライン間でトラッキングが途切れてしまうことがあげられる。特に広坂交差点ではバスなどの大型車の車両に隠れてしまいカウンターラインと交差しない位置から追跡対象の車両が現れるといったことが多く発生したため比較的計測精度が低くなっていた。このような場合、よりカウントの精度を上げるために設置するカウンターラインの本数を増やすといったことが考えられるが、これは設定作業のコストが増加することや、撮影するカメラの位置や交差点の状況に依存するため一般の交差点に対応するのは難しい。そこで今後の展望としては途中で途切れてしまった軌道の復元や移動方向の推定を行うことで計測精度を改善していきたい。

## 6. まとめ

本研究では、MOTを用いた交通量調査におけるカウンターラインの設定作業の一部の自動化を実現した。提案手法では人手によりカウンターラインを設定するエリアを与えた後に、ランダムサーチで探索を行い各エリアにおいて適切なカウンターラインの候補の選定を行い、それらの組合せでエリア間のカウント数が実測値に最も近い組みを採用した。提案手法の汎用性を示すため金沢市内の2カ所の交差点で撮影した動画を対象にnode-moving-things-trackerとDeepSORTの2つのトラッキング手法で車両のトラッキングを行い、計測精度の比較を行った。また、人手によるカウンターラインの設定を5人にそれぞれ5回ずつ行ってもらい、一番良かった結果を採用し、提案手法との比較を行った。その結果、いずれの交差点においても人手での設定以上の精度が出ることを確認した。

今後の課題として、現在カウンターラインの設置範囲の設定を手で行っているため完全な自動化の実現が望まれる。また、今回は比較的容易に使用できるランダムサーチを利用したが、より効率的な探索手法を使用した場合の処理時間の比較検証を行っていく必要がある。さらに、現状では2カ所の交差点でしか検証できておらず、短期計測における正解データでの評価にとどまっているため、長期計測での評価を行うとともに、異なる交差点や時間帯で変わる流量、天候の変化が計測に与える影響についても検証していきたい。

## 参考文献

- [1] 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] 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] 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] OpenDataCam 3.0.2 – An open source tool to quantify the world (GitHub), available from (<https://github.com/opendatacam/opendatacam>) (accessed 2021-11-24).
- [5] node-moving-things-tracker (GitHub), available from (<https://github.com/opendatacam/node-moving-things-tracker>) (accessed 2021-11-24).
- [6] Yolov5 + Deep Sort with PyTorch (GitHub), available from (<https://github.com/mikel-brostrom/Yolov5-DeepSort-Pytorch>) (accessed 2022-03-28).
- [7] Scorer Traffic Counter, available from (<https://www.scorer.jp/products/scorer-traffic-counter>) (accessed 2021-11-24).
- [8] 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).
- [9] 簡単らくらく通行量分析, 入手先 (<https://www.chess-inc.com/odpt/>) (参照 2021-11-24).
- [10] Zhang, Y., Wang, C., Wang, X., Zeng, W. and Liu, W.: FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking, *Int. J. Comput., Vision* 129, 11 (2021).
- [11] Chu, P., Wang, J., You, Q., Ling, H. and Liu, Z.: TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking, ArXiv, abs/2104.00194 (accessed 2021-11-24).
- [12] Alexey, B., Chien-Yao, W. and Hong-yuan, L.: YOLOv4: Optimal Speed and Accuracy of Object Detection, available from (<https://arxiv.org/abs/2004.10934>) (accessed 2021-11-24).
- [13] Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B.: Simple online and realtime tracking, *2016 IEEE International Conference on Image Processing (ICIP)* (2016).
- [14] Wojke, N., Bewley, A. and Paulus, D.: Simple online and realtime tracking with a deep association metric, *2017 IEEE International Conference on Image Processing (ICIP)* (2017).
- [15] Dulbadrakh, T., 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, 情報処理学会第83回全国大会, 7Q-07, 情報処理学会 (2021).
- [16] 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 2021-11-24).
- [17] Bochinski, E., Eiselein, V. and Sikora, T.: High-Speed tracking-by-detection without using image information, *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2017).
- [18] Bochinski, E., Senst, T. and Sikora, T.: Extending IOU Based Multi-Object Tracking by Visual Information, *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2018).
- [19] Yang, H., Zhang, Y., Zhang, Y., Meng, H., Li, S. and Dai, X.: A Fast Vehicle Counting and Traffic Volume Estimation Method Based on Convolutional Neural Network, *IEEE Access*, Vol.9, pp.150522–150531 (2021).
- [20] Andres, O. and Felipe, T.: Countor: Count Without Bells and Whistles, *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020).

- [21] OpenDataCam 2.0 - User Interface Walkthrough, available from (<https://vimeo.com/346340651>) (accessed 2021-11-25).



渡辺 魁 (学生会員)

2022 年金沢工業大学工学部情報工学科卒業。現在、金沢工業大学大学院工学研究科情報工学専攻博士前期課程在学。交通流シミュレーションに関する研究に従事。



中野 勝章 (学生会員)

2022 年金沢工業大学工学部情報工学科卒業。現在、金沢工業大学大学院工学研究科情報工学専攻博士前期課程在学。深層学習による動画像認識に関する研究に従事。



中沢 実 (正会員)

1993 年金沢工業大学大学院工学研究科情報工学専攻博士前期課程修了。同年(株)富士通研究所入社。1996 年金沢工業大学教授となり、現在に至る。博士(工学)。自律分散システム、ロボティクス、信号処理に関する研究に従事。2016 年情報処理学会山下記念研究賞受賞。電子情報通信学会、IEEE 各会員。



長沼 要

1993 年北海道大学大学院工学研究科機械工学専攻修了。同年日産自動車株式会社入社、総合研究所にてエンジン制御に関する研究開発に従事。その後低公害車両開発会社立ち上げや多くの環境負荷低減技術開発プロジェクトに関与。2016 年金沢工業大学教授。博士(工学)、日本機械学会、自動車技術会、SAE 各会員。