

ドライブレコーダを用いた道路環境解析のための
車両および道路附属物の位置・軌跡推定

Position and Trajectory Estimation of Vehicles and Roadside Equipment
for Road Environment Analysis Using Dashboard Cameras

川崎 敦史¹⁾ 高橋 俊裕¹⁾ 渡辺 友樹¹⁾
Atsushi Kawasaki Toshihiro Takahashi Tomoki Watanabe

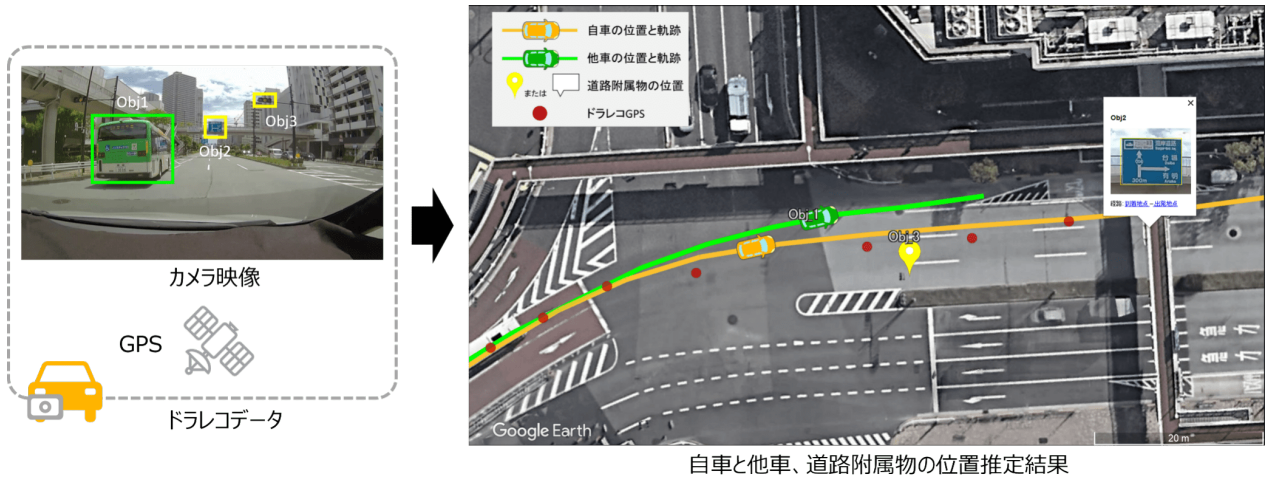


図 1: 提案法で推定した自車・他車・道路附属物の位置・軌跡を Google Earth [11] 上にマッピングした結果.

1 はじめに

近年、道路上の交通状況やインフラ設備のデジタル化が求められている [1]. 交通状況のデジタル情報では、道路上の移動体の動きを広域で把握することで、交通・流通の効率化や交通事故状況の解析等に役立つ. 道路周辺のインフラ設備のデジタル情報では、信号機や交通標識等の道路附属物の位置を把握することで、インフラの設計・施工・管理・保守といった業務の効率化に役立つ.

効率化の具体例として、事故状況解析の用途では、過失割合算出等の保険業務に必要な事故状況図の作成にあたり、熟練者でも困難であった、自車と衝突物の地図上の動きを算出する技術が求められている [2]. また、道路附属物の保守・点検の用途では、附属物の位置情報と点検内容を作業者が手作業で紐づけていたが、人手や費用が大きな負担となっており、負担軽減のために附属物の地図上の位置を算出する技術が求められている [3].

従来使われている方法として、道路上の移動体や附属物の位置や動きは、カメラや RTK-GPS, LiDAR を搭載した専用車両 (Mobile Mapping System など) を用いることで取得できる. RTK-GPS では自車位置を, LiDAR では移動体を含む周辺環境の対象の位置を, それぞれ数センチ以内の誤差で計測できる. しかし, これらのセンサは高価で, 一般の車両には搭載されていないため, いつ発生するかわからない事故状況解析に用いることは困難であり, 道路附属物の保守点検の用途においても保守コストの増加を招くため好ましくない. 一方で, ドライブレコーダ (以下, ドラレコ) は安価かつ取り扱いが容易であり, 交通事故時の証拠記録のために広く普及している. そこで, ドラレコで取得可能な映像と GPS のみ

を用いて, 上記の用途で求められる自車位置推定と周辺環境認識の実現を目指す (図 1).

しかし, 本目的の実現には大きく 2 つの技術的課題がある. 1 つ目は自車位置の計測精度である. RTK-GPS は, 位置が既知の基準局を含む複数の受信機で位置を計測する方式をとっており, 誤差 2cm 程度の高精度な位置計測が可能である. しかし, ドラレコの GPS は 1 台の受信機で位置を計測しているため, 1~10m ほどの誤差が頻繁に生じてしまう. また, 映像から自車の動きを推定する SLAM (Simultaneous Localization and Mapping) または SfM (Structure from Motion) [4] と GPS を組み合わせた手法 [5, 6] も提案されているが, 高精度な GPS や, GPS 誤差を低く見積もったシミュレーションを用いており, 誤差が大きいドラレコ GPS では, 逆に悪化してしまう.

2 つ目の課題は, 周辺環境の対象物, 特に移動体の計測精度である. SLAM / SfM は, カメラの位置姿勢と画像上の対応点の 3 次元位置を同時に最適化する手法であり, 周辺環境の 3 次元情報も算出できる. しかし, 単眼カメラ画像による SLAM / SfM では原理的に移動体の 3 次元位置を算出できない. この問題に対し, 昨今の深層学習関連の技術の発達により, 主に 2 つの解法が提案されている. カメラ画像から DNN (Deep Neural Networks) で, 3 次元の bounding box (3D-bbox) を推定する方法 [7] と, 画素単位の奥行きを推定する方法である. 特に後者では, LiDAR 等で計測した奥行きを真値として学習する手法 [8] や, 教師なしで学習する手法 [9], 大量の学習データを用いた汎用的な奥行き推定 DNN [10] も提案されている. しかし, どの手法を用いても, 単眼画像から移動体の 3 次元位置を高精度に推定

1) (株) 東芝, Toshiba Corporation.

することは難しく、また自転車位置推定の誤差も加わり、移動体の位置を時系列で上面図上にマッピングすると、バラついた不安定な軌跡になってしまう。

本論文では、以上の技術的課題を解決するために、GPS 誤差が大きいドラレコを用いて、地図上の自転車軌跡を推定する技術と、周辺の移動体および道路附属物の位置・軌跡を推定する技術を開発した。自転車の軌跡推定では、GPS の低精度な絶対位置と映像からの高精度な相対運動推定を統合し、高精度かつ安定した地図上の軌跡推定を実現する。周辺の移動体や道路附属物の位置・軌跡推定では、三次元幾何と深層学習による画像からの物体検出、奥行き推定、時系列補正により、高精度かつ定性的に安定した推定を実現する。図 1 に、ドラレコデータから、自転車・他車・道路附属物の地図上の位置・軌跡を推定した結果を示す。これらの結果を Google Earth [11] と連携することで、地図や航空写真と重ね合わせて効率的に閲覧・管理できる。

提案法における自転車軌跡の推定では、安定性と精度を向上させるために、ドラレコの特性を考慮した 3 つのステップによる手法を提案する。1. DNN による安定した相対運動推定、2. 少数の画像上の対応点を用いた SfM ベースの最適化による補正、3. GPS を用いた位置姿勢の実スケール化と地図上への位置合わせからなる。相対的な位置姿勢の推定を画像のみで行うことで、GPS 誤差に影響されない自転車の相対的な動きを推定する。さらに、補正に用いる対応点を少数にすることで、処理時間を軽減させた。本手法はオフライン処理を想定しているが、処理時間の削減により、作業者の待ち時間減といった効果がある。

周辺対象の位置推定では、静止体と移動体の 2 種類を扱い、それぞれに対して別手法を用意する。道路附属物を始めとする静止体は、SfM で求めた周辺環境の 3 次元情報を用いる。一方、原理的に SfM では算出できない移動体の 3 次元位置は、奥行き推定 DNN で推定し、上面図上にマッピングして軌跡を作成する。その後、新たに提案する軌跡補正 DNN により、軌跡誤差を補正し、滑らかな軌跡推定を実現する。DNN の学習では、奥行き推定等で作成した軌跡の誤差をモデル化し、真値軌跡に対してモデルに基づく誤差を付与するデータ拡張を通じて、多様な入力軌跡を生成し、高精度化を実現した。

実験は市販のドライブレコーダで行った。提案法で推定した自転車の位置姿勢、他車両や歩行者などの移動体の位置姿勢、道路附属物の位置を、それぞれ評価した。既存手法と比較して、自転車の位置姿勢は安定かつ高精度に推定できることを示し、移動体についても高精度かつ定性的に滑らか軌跡を推定できることを示した。

本論文の Contributions を以下にまとめる：

- 専用車を使わずに、安価かつ取り扱いが容易なドラレコを用いて、自転車や周辺車両、道路附属物の位置・軌跡を高精度に推定する一連のアルゴリズムを提案する。市販のドラレコを用いた実験では、既存手法よりも高精度に推定できることを示した。
- GPS 誤差が大きいドラレコデータの特性を考慮して、高精度かつ安定した自転車位置を推定する。また、補正には少数の対応点を使うことで、処理時間を軽減させる。
- SLAM / SfM では原理的に算出できない移動体の位

置を、DNN による奥行き推定と時系列補正を用いることで、高精度かつ時系列的に安定した軌跡となるように推定する。

2 全体の処理概要

図 2 に本論文で提案する自転車や移動体、道路附属物の位置・軌跡推定手法の全体像を示す。入力は、タイムスタンプ付きの映像と GPS を含むドラレコデータである。出力は、自転車の時系列の緯度経度と向き、移動体の時系列の緯度経度と向き、道路附属物の緯度経度である。

はじめに、自転車の位置姿勢推定 (図中の橙枠内) について説明する。カメラ映像から DNN を用いて自転車の相対的な位置姿勢を推定し、これを初期値とする。次に、画像上の対応点を用いた SfM ベースの最適化で、自転車位置を補正する。最後に、GPS で位置姿勢の実スケール化と地図への位置合わせを行う。以上の 3 ステップで地図上の自転車位置と向きを算出する。

次に、道路附属物等の静止体の位置推定について説明する。まず、off-the-shelf の物体認識 DNN [12] で、道路附属物 (信号機や標識、青看板など) を検出する。次に、自転車位置推定の SfM で使用した画像上の対応点の中から、検出矩形に含まれるものを抽出する。SfM では各対応点の 3 次元位置を推定しているため、抽出した対応点の 3 次元点を参照し、この 3 次元座標を道路附属物の地図上の位置とする。

最後に、移動体の位置推定 (図中の緑枠内) について説明する。まず、物体認識 DNN [12] で、移動体 (車、歩行者など) を検出・追跡する。次に、off-the-shelf の奥行き推定 DNN で、各画像における画素単位の奥行きを推定する。ここで、奥行き推定 DNN は、ネットワーク構造に [13] を使用し、LiDAR で計測した奥行きを真値として学習した。検出矩形内の画素に該当する奥行きから、移動体とカメラ間の相対的な位置関係を算出し、推定した自転車位置を基準に地図上にマッピングすることで、移動体軌跡の初期値を算出する。最後に、初期軌跡を軌跡補正 DNN で補正することで、最終的な移動体の位置と向きを算出する。

3 自転車の位置・軌跡推定

本章では、ドラレコデータの特性を考慮した、安定かつ高精度な自転車位置推定手法を提案する。ドラレコデータの特性として、GPS の精度が低く、1~10m ほどの誤差が生じることや、画像と GPS の同期が完璧ではなく、ミリ秒単位のズレが生じることが挙げられる。そのため、既存手法 [5, 6] のように画像上の特徴点に GPS を加えた同時最適化で位置推定を行うと、推定に悪影響を与えてしまう。そこで、画像のみで相対的な位置姿勢を推定し、GPS は実スケール化と絶対座標への位置合わせのみに使用することにした。また、単眼カメラの SLAM / SfM の課題である安定性の低さを解決するために、高精度ではないが安定性が高い DNN で推定した位置姿勢を初期値とし、高精度に推定可能な SfM ベースの最適化で補正する手法を提案する。さらに、補正に用いる対応点を少数にすることで、処理時間の軽減も行う。

3.1 DNN による相対的な位置姿勢推定

位置姿勢を推定する DNN (以下、位置姿勢 DNN) では、2 枚以上の連続する画像を入力として、画像間の相対的な位置姿勢を出力する。本論文では、教師なし学習による Off-the-shelf の位置姿勢 DNN [9] を使用する。

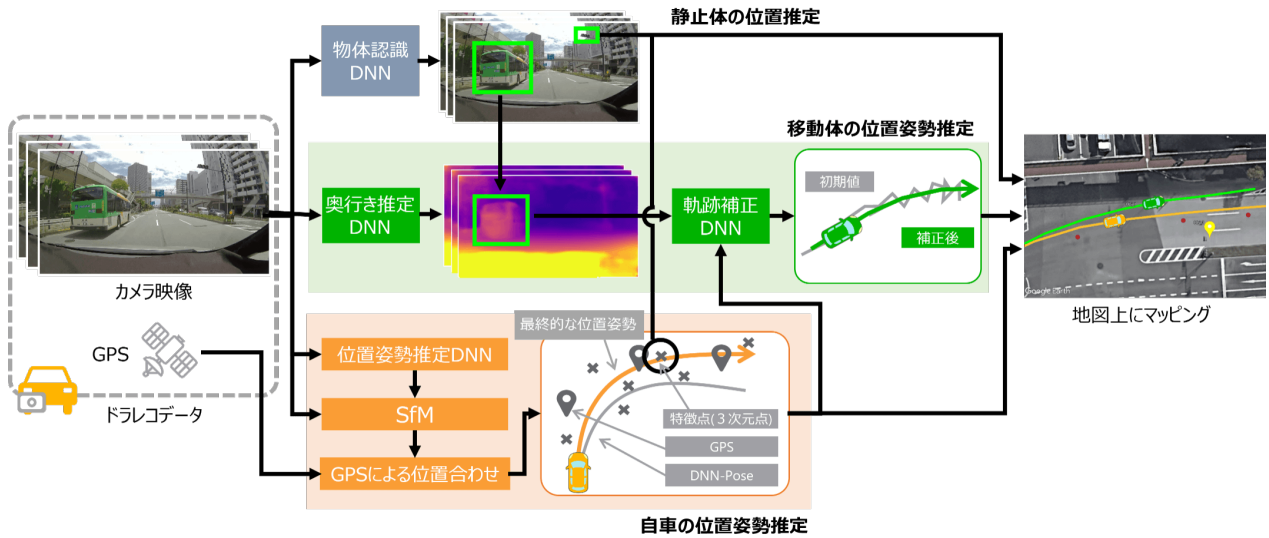


図 2: 自転車や移動体、道路附属物の位置・軌跡推定の処理概要。

教師なし学習では、奥行きマップを推定する奥行き推定 DNN と、位置姿勢 DNN から構成される。元画像のピクセルに対応した奥行きマップのピクセルは、位置姿勢により前後のフレームへ写像でき、実画像と写像した画像（合成画像）の間の輝度損失を最適化することで、教師なし学習を実現する。提案法では、こうして学習した位置姿勢 DNN のみを使用する（図 2 中の奥行き推定 DNN と、教師なし学習で使った奥行き推定 DNN は異なる）。推定結果の特徴として、位置姿勢 DNN は大量の走行データから学習しているため、画像上の特徴が少ないシーンでも車の動きから大きく外れるような破綻した結果が生じにくい。ただし、教師なし学習のため高精度ではなく、後段で補正を行う必要がある。

3.2 SfM による位置姿勢の補正

基本的な処理の流れは、通常の SfM と同じで、特徴点の検出、特徴点のマッチング、三角測量による 3 次元点の初期値推定、Bundle Adjustment (BA) による全体最適化からなる。本論文では、特徴点の検出に、ロバストかつ高速なアルゴリズムである AKAZE [14] を使用する。また、移動体の影響を軽減させるために、物体認識 DNN で得られた車や歩行者などの物体上の特徴点はマスクした。3 次元点の初期値推定では、位置姿勢 DNN で得られた位置姿勢と対応点から、三角測量により求める。全体最適化では、三角測量で得られた 3 次元点、位置姿勢 DNN で得られたカメラの位置姿勢を用いて、再投影誤差（特徴点の座標と 3 次元点を画像に投影した座標の距離）が最小になるように、カメラの位置姿勢と空間上の 3 次元点を最適化する。安定性が高い DNN で推定した位置姿勢を初期値としているため、SfM 後の結果も安定して得られる。

さらに、本論文では、対応点間引きによる少数の対応点での全体最適化を提案する。一般的に、対応点が増えると BA の推定変数が増えるため処理時間が増大する。自転車位置姿勢の補正目的であれば、過剰な対応点は不要のため、必要最低限の対応点に間引き、処理時間の削減を狙う。位置姿勢推定の精度に寄与する対応点として、対応点のフローが大きいもの、観測フレーム数が多いものが挙げられる。位置姿勢は見え方の変化から推定

するため、フローの大きさは最も重要な要素である。また、観測フレーム数が多いほど、フレーム間の対応付けが増え、スケールドリフトを防ぐ効果がある。提案法では、これらの対応点を維持しつつ、最適化への影響が軽微な対応点を除去する。

Algorithm 1 に対応点間引きの処理フローを示す。はじめに以下の 4 つの閾値を用意する。

- O_{th} : 対応点の観測フレーム数のしきい値
- F_{th} : 対応点フローのしきい値
- S_{th} : 1 フレーム内に必要な特徴点数のしきい値
- A_{th} : 総対応点数のしきい値

最初の 2 つは、ある対応点における観測したフレーム数とフローに対するしきい値で、これらを下回るものが削除候補となる。残りの 2 つは、最適化時に必要最低限な 1 フレーム内の特徴点数と、総対応点数である。処理フローを説明する。まず、各対応点について着目し、ある対応点の観測フレーム数とフローが、それぞれのしきい値以上である場合、最適化に有効な対応点とみなし保持する。一方、しきい値をみたさない場合、2 つのチェックを行う。1 つ目は、この対応点の削除により、1 フレーム内の特徴点数がしきい値以下になるフレームが、1 つでも存在する場合、最適化に不利になるため、この対応点は削除しない。2 つ目は、この対応点の削除により、総対応点数がしきい値以下になる場合、これ以上の対応点は削除できないため、間引き処理を終了する。これらのチェックをパスした対応点は、最適化への影響は軽微として削除する。これらの処理を全対応点について行った後、観測フレーム数と対応点フローのしきい値を増やす。以上の処理を複数回繰り返すことで、最適化時に必要な 1 フレーム内の特徴点数と総対応点数をみたしつつ、不要な対応点を手続的に除去できる。

3.3 GPS による実スケール化と絶対座標変換

SfM で得られる位置姿勢は単眼カメラを用いた推定なので、相対的かつスケール（移動距離の単位）は不定である。そのため、GPS を用いた実スケール化と地図上の絶対座標への変換を行う。まず、GPS の経緯度は、最初に正常に観測できた経緯度を座標原点としたメルカ

Algorithm 1 SfM の対応点間引き

```

1: for  $n = 1 \dots N_{do}$ 
2:   for  $i = 1 \dots$  総対応点数 do
3:     if  $i$  番目の対応点の観測フレーム数  $< F_{th}$  then
4:       continue
5:     else if  $i$  番目の対応点のフロー  $< F_{th}$  then
6:       continue
7:     else if 1 フレーム内の特徴点数  $< S_{th}$  then
8:       continue
9:     else if 総対応点数  $< A_{th}$  then
10:      return
11:    end if
12:     $i$  番目の対応点を削除
13:  end for
14:   $O_{th} \leftarrow O_{th} + 1$ 
15:   $F_{th} \leftarrow F_{th} + 1$ 
16: end for

```

トル展開で水平面に展開する [15]. 次に, SfM による時系列の自転車位置を, メルカトル展開された GPS の位置座標に近づかせる相似変換パラメータ (スケール s ・回転 \mathbf{R} ・並進 \mathbf{t}) を Horn の手法 [16] で推定する. 推定パラメータに基づき, 以下の式で, SfM 後の自転車位置 \mathbf{x}_{ego}^t をメルカトル展開された GPS 座標系に変換する:

$$\hat{\mathbf{x}}_{ego}^t = s\mathbf{R}\mathbf{x}_{ego}^t + \mathbf{t} \quad (1)$$

最後に, GPS 座標系の自転車位置をメルカトル展開の逆変換で経緯度に変換する.

以上の方法により, カメラで推定した高精度な軌跡形状を変化させずに実スケール化と絶対座標変換ができる. また, Horn の手法ではシーケンス全体を用いて変換パラメータを推定するため, ドラレコ GPS の誤差影響は最小限に緩和される.

4 道路附属物の位置推定

本章では, 信号機や看板などの道路附属物の位置推定について説明する. 道路附属物は静止体のため, SfM で算出した周辺環境の 3 次元情報を使用できる. SfM では静止体であることを前提として 3 次元情報を算出するため, 後述する移動体位置推定の枠組みを使用するよりも, 手法として適している.

まず, 物体認識 DNN で附属物を矩形で検出し, 矩形内に存在する SfM で使用した特徴点に対応する 3 次元点を取得する. ここで, 3 次元点は GPS により実スケール化され絶対座標に変換されたものを用いる. そして, 取得した複数の 3 次元点における x, y, z 軸のそれぞれの中央値を附属物の位置とした.

5 移動体の位置・軌跡推定

本章では, 他車両や歩行者などの移動体の位置・軌跡推定について説明する. 移動体の 3 次元位置は, 原理的に SfM では算出できないため, 物体認識 DNN による画像上の位置と, 奥行き推定 DNN によるカメラからの相対的な距離で算出する. 各時刻の相対的な位置と, 自転車位置推定結果から, 移動体の時系列の位置 (軌跡) を上面図上にマッピングできるが, マッピングした軌跡は, 奥行き推定, 自転車位置推定等の誤差により, 不安定にな

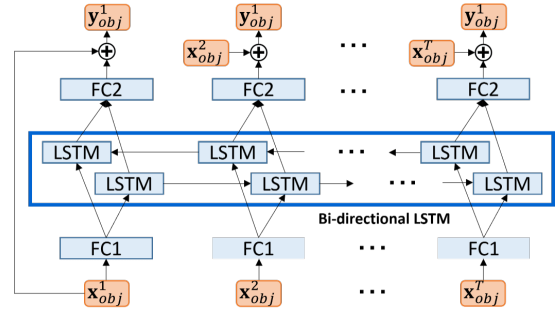


図 3: 軌跡補正 DNN のネットワーク構成.

る. そこで, マッピングした軌跡を入力として, 軌跡誤差を補正する軌跡補正 DNN を提案する.

5.1 軌跡補正 DNN の入力軌跡

軌跡補正 DNN の入力, 2 次元平面上の軌跡である. まず, カメラからの相対的な移動体の 3 次元位置を, 物体認識結果と対象までの奥行きを組み合わせて算出する. 具体的には, 検出矩形の下辺中点の座標に, 矩形内の奥行き 20 パーセント値を奥行きとして与え, 既知のカメラの内部パラメータで逆投影する. パーセント値を用いたのは, 物体同士のオクルージョン等を考慮したためである. 次に, 各時刻の移動体の 3 次元位置を, 自転車位置推定結果を用いて, 特定フレームのカメラ座標系に変換する. ここで特定のフレームは, 自転車と移動体の距離が最も近いフレームとした. 座標変換した 3 次元点を x - z 平面に投影した軌跡を $\{\mathbf{x}_{obj}^1, \dots, \mathbf{x}_{obj}^T\}$ とし, 軌跡補正 DNN の入力とする.

5.2 軌跡補正 DNN

軌跡補正 DNN のネットワーク構造を図 3 に示す. 軌跡補正 DNN は, 自動運転向けの観測軌跡から将来軌跡を予測する技術 [17] を応用し, 誤差を含む軌跡から補正軌跡を出力するように設計した. 軌跡補正 DNN は, 時系列の入力軌跡 $\{\mathbf{x}_{obj}^1, \dots, \mathbf{x}_{obj}^T\}$ を FC 層と Bi-directional LSTM [18] を通じて, 補正軌跡 $\{\mathbf{y}_{obj}^1, \dots, \mathbf{y}_{obj}^T\}$ を出力する. 本手法はオフライン処理を前提としているので, 順方向だけでなく逆方向の時刻ステップを考慮できる Bi-directional LSTM を使用した. 入力軌跡の誤差の特徴として, 移動体の位置誤差は遠方になるほど大きくなる. そのため, 単方向の LSTM では, 追い越し車両 (誤差小 \rightarrow 大) は対応できるが, 近づく車両 (誤差大 \rightarrow 小) の対応は難しい. その点, 双方向の時刻情報を使用できる Bi-directional LSTM は本目的に適している. 出力部は, 入力との和をとる skip connection を導入し, 学習が必要なネットワークの最終層は補正量を出力する構成にした. 軌跡の座標値よりも, 補正量の値域は限定的なので, DNN の学習を容易にする効果が期待される.

移動体の向きは, 隣接時刻の位置差分により算出する. このとき, 移動量が一定値以下の場合, 差分を取る時刻幅を増やすことで, 結果を安定化させた.

5.3 学習データ

軌跡補正 DNN を学習するための公開データセットは存在しないため, 自前で作成した. 軌跡データの真値は, LiDAR 等で計測した 3 次元点群に 3D-bbox を教示することで取得した. 入力の誤差軌跡は, 実映像に対して物体認識 + 奥行き推定等の処理をすることで算出でき

る。ただし、3D-bbox の教示はコストが大きく、用意できる学習データは限定的である、そこで、入力軌跡の誤差をモデル化し、モデルに基づく多様な誤差を付与するデータ拡張をすることで、学習データの増強を行う。

入力軌跡の誤差のモデル化にあたり、誤差傾向を再現する必要がある。物体認識+奥行き推定で算出した移動体の 3次元位置は、自転車から遠いほど誤差は大きく、遠方の対象は手前に推定されやすいといった傾向を持っていた。そこで、自転車を中心とする距離 (10m ごと) と角度 (30 度ごと) で区切った領域を作成し、領域ごとに誤差の分布をモデル化した。各領域に真値位置が属する移動体の位置誤差から統計量 (誤差平均+分散) を算出する。以上により作成した誤差モデルに従い、真値軌跡にランダムで誤差を付与しながら学習を行う。

6 実験

6.1 データセット

実験で使用するデータセットについて説明する。GPS と映像を含むドラレコデータ、さらには位置真値を含む公開データセットは存在しないため、自前で用意した。実験用の車両に、映像と GPS を取得できる市販のドラレコ、高精度な位置情報を取得できる RTK-GPS, LiDAR, ステレオカメラ等を搭載し、公道でデータを収集した。RTK-GPS, LiDAR, ステレオカメラは同期が取れるように設計している。これらのデバイスと市販のドラレコの同期は、カメラ画像上で信号灯が切り替わるイベント等を目印に目視で行った。ドラレコは、FullHD の映像を 27fps で、GPS は 2Hz で取得できる。一方、RTK-GPS, LiDAR は 10Hz で取得できる。

評価データは、15~30 秒の動画長の 23 シーンで、各シーンは交差点を含み、移動体が多い都心部で撮影した。14 シーンは日中、9 シーンは夜間のデータである。

自転車位置の真値は RTK-GPS を、方位の真値は RTK-GPS から取得できる速度ベクトルを使用した。移動体の真値は、LiDAR で計測した点群から自転車との相対的な位置や向きを示す 3D-bbox を算出し、RTK-GPS を基準に地図上にマッピングすることで作成した。道路附属物の位置真値は、衛星画像から目視で位置を確認し、経緯度を取得した。

移動体の評価対象は、車・バイク・人・自転車で、対象数はそれぞれ、123, 9, 24, 9 である。1 フレームでも 10m 以内に近づいた 50m 以内の対象を評価した。移動速度によって傾向が異なったため、車とバイク、人と自転車に分けて評価する。道路附属物の評価対象は、道路上の青看板・信号機・標識で、計 32 個を用いた。

6.2 評価向け前処理

評価にあたり、推定した軌跡に対する前処理を検討する。前述した通りドラレコの GPS は 1~10m ほどの誤差が生じる。この誤差には 2 つの傾向がある。1 つ目は時系列的にランダムに生じる誤差、2 つ目は真値位置から定常的に生じる誤差である。ドラレコの GPS は、1 台の受信機で複数の衛星からの電波を受信し、各衛星と受信機間の距離からの三角測量で絶対位置を算出している。そのため、電波の受信に定常的な誤差 (大気遅延や、建物等の反射によるマルチパスなど) が生じると、地図上の位置にも定常的な誤差が生じる。

提案法は、ランダムに生じる誤差は補正できるが、定常的な誤差は補正できない。そのため、最終的な応用例

表 1: 実応用における位置誤差 [m] のまとめ。

	自転車	移動体 (車・バイク)	移動体 (人・自転車)	道路 附属物
GPS	2.5	-	-	-
提案法	1.5	6.3	2.7	2.7

である事故映像解析や道路附属物点検では、地図 API から取得できる車線情報または手動による、推定軌跡全体を回転+並進移動させた補正を想定している。

補正後の位置・軌跡を評価するために、今回は RTK-GPS から得られる真値軌跡に、推定した自転車軌跡を回転+平行移動により近づかせることで補正する。具体的には、同一時刻の各軌跡の位置誤差が最小となるように、回転+平行移動の変換パラメータを算出する [16]。このパラメータで、自転車軌跡・移動体軌跡・附属物位置を変換することで、地図に合った軌跡・位置に補正する。ここで、推定軌跡の曲がり具合などの形状は、この変換で変化しないことは注意されたい。

図 4 に GPS の定常誤差の補正前後の結果を示す。図中のピンク丸はオリジナルのドラレコ GPS を指し、真値の青線と比較すると定常的な誤差が生じている。オリジナルの GPS を用いて推定した自転車と他車の軌跡も、定常的な誤差が生じており、補正が必要であることがわかる。実応用における補正を再現した自転車と他車の軌跡を橙線と緑線で示す。1 つの回転+平行移動をするだけで、シーン全体の推定軌跡と映像の一貫性が取れるようになっていく。以降、定性評価では補正後の軌跡を示し、定量評価では補正前と補正後の双方の誤差を示す。

6.3 実応用における定量評価

提案法の要素ごとの評価をする前に、最終的な応用例である事故映像解析や道路附属物点検に必要な、自転車・移動体・道路附属物の位置誤差を表 1 にまとめた。位置誤差は、GPS 補正後の推定値と真値とのユークリッド距離である。各シーケンスで動画長が異なるため、シーケンスごとに誤差平均を算出したのち、全シーケンスの平均値を算出した。自転車位置誤差については、GPS と比べて提案法により改善していることがわかる。移動体や道路附属物の位置は、自転車位置を基準に算出しているため、それらの誤差は自転車位置誤差よりも大きくなった。以降、提案法の各要素について、詳細な評価を行う。

6.4 自転車位置推定における評価

6.4.1 定性評価

図 4, 5, 6 に、自転車位置の推定結果を橙軌跡、ドラレコ GPS を赤丸、真値を青軌跡で示す。また、車アイコンは各図の左上画像の時刻の車両位置である。ドラレコ GPS はバタつきが大きいのに対し、推定した軌跡は滑らかで真値に近いことがわかる。特に、図 6 の左折時に一時停止をするシーンにおいて、停止時のドラレコ GPS の分散は大きいだが、推定軌跡は安定している。提案法では軌跡形状の推定に画像のみを用いているため、ドラレコ GPS の大きい誤差の影響を受けずに推定できた。

6.4.2 定量評価

自転車の地図上の位置・向きの誤差を評価する。比較手法には以下を用いる：

- Colmap [19]・・・OpenSource の 3 次元再構成ツールでカメラ画像から SfM を行う。BA 前の初期値は、5 点法や PnP 問題などの幾何的解法を使用。実

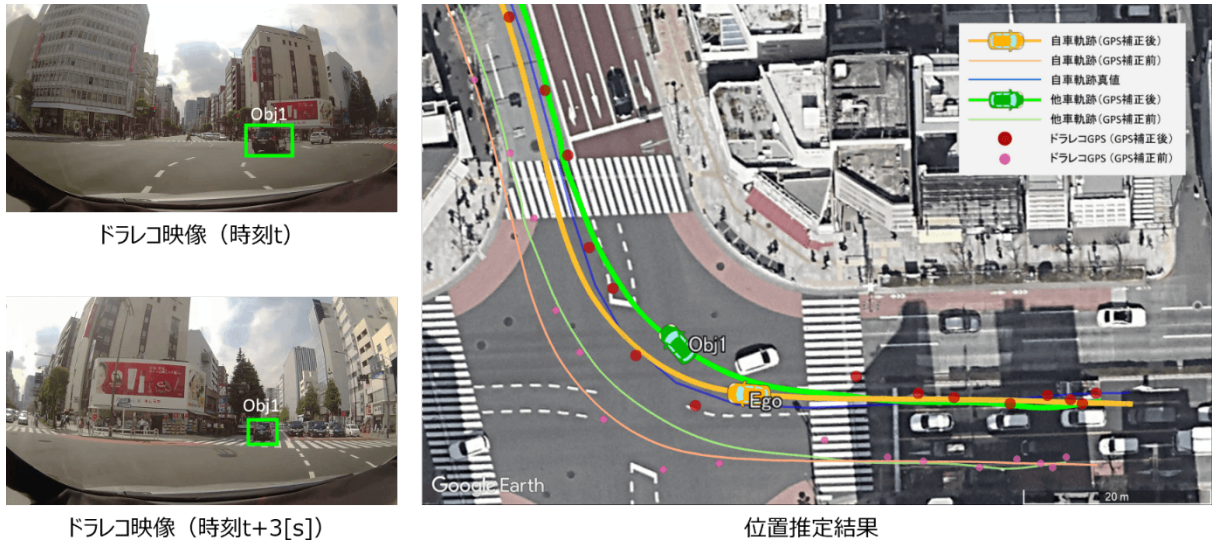


図 4: 自転車・他車両の軌跡推定結果. GPS の定常誤差の補正前後の比較.

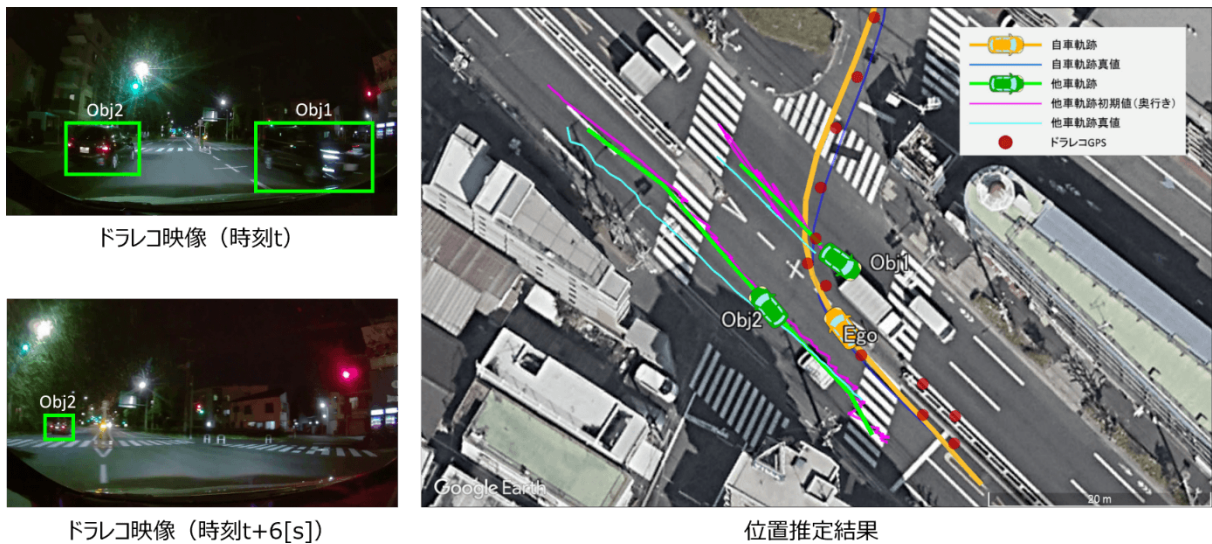


図 5: 自転車・他車両の軌跡推定結果. 他車両軌跡の補正前後の比較.

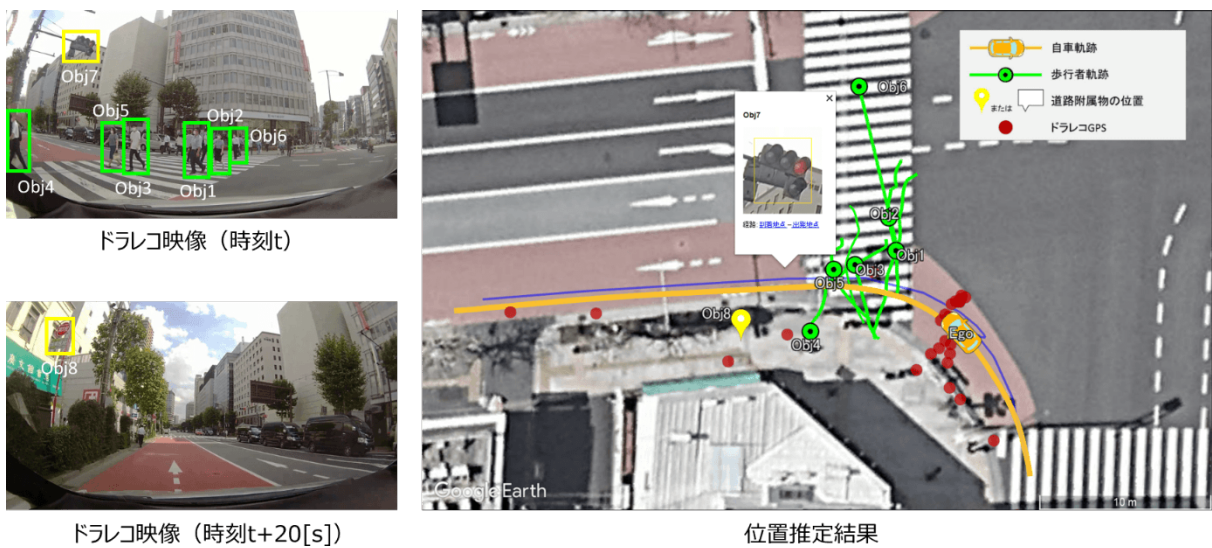


図 6: 自転車・歩行者・道路附属物の位置・軌跡推定結果.

スケール化は提案法同様に GPS を使用. 移動体上の特徴点マスクも同様に使用.

- 同時最適化 [5]・・・画像上の特徴点と GPS の絶対位置を用いた同時最適化により位置推定する手法.
- DNN-Pose [9]・・・提案法における SfM による位置補正を除き, DNN による位置姿勢推定と GPS による実スケール化のみを行う.

提案法と比較手法の位置誤差と向き誤差を表 2 に示す. GPS 補正後の数値では, 提案法が位置・向きともに最も精度が高い. 一方, 補正前の数値から有意差は見られなかった. これは, GPS の定常的な誤差が支配的で, 各手法の効果が見えなくなっているためである.

各手法と比較すると, 同時最適化手法 [5] の位置誤差は最も大きかった. これは, ドラレコ GPS の誤差が大きいため, 同時最適化による相乗効果が得られず, 逆に不安定な結果になったためである. DNN-Pose [9] と提案法の誤差の差分は, SfM による位置補正の効果を示している. DNN-Pose は隣接フレームしか見ておらず, 誤差が蓄積して大きくなっているが, SfM による全体最適化で約 4 割の位置誤差を削減できた. Colmap [19] と比較すると, どちらも SfM による最適化を行っているが, わずかに提案法の結果が優れていた. 5 点法などの幾何的手法と比べ, DNN で推定した初期値が夜間等のシーンでも精度が良く, SfM 後の結果も高精度になったと考える. また, Colmap [19] では, 23 シーン 20 シーンしか SfM が成功せず, すべてのカメラ位置を出力できなかった. 一方, 提案法は DNN による初期値を用いたことで, 安定して全シーンのカメラ位置を出力できた. 以上により, 提案法は従来法と比べて, 安定かつ高精度な自転車位置推定を実現したと言える.

次に, SfM における対応点の間引き処理の効果を評価する. 実験では, システム全体を CPU 処理 (Intel Xeon Gold 5317 CPU @3.0GHz) のみで動かした. 表 3 に間引き処理がある場合/ない場合の位置誤差と自転車位置推定処理の処理時間を示す. 参考に SfM ベースの手法である Colmap [19] の結果も示す. 動画長の基準を合わせるために, 15 秒の動画 (9 シーン) で評価した. 対応点の間引きあり/なしを比べると, 位置精度を落とさずに, 処理時間を約 30%削減できた. これは, 提案法による対応点の間引き処理の効果を示している. また, Colmap [19] と比較すると, 約 9%の処理時間である. これは対応付けの探索フレーム数や最適化の繰り返し数が異なることが起因している. ただし, あくまで Colmap は汎用的な 3 次元再構成ツールであり, 提案法は車載カメラ映像に特化した手法であることに注意されたい. システム全体の処理時間も表 3 に記載する. 自転車位置推定の処理時間は全体の約半分であった.

6.5 移動体位置推定における評価

6.5.1 定性評価

図 4, 5 に他車両の推定軌跡を, 図 6 に歩行者の推定軌跡を, 緑軌跡で示す. 特に図 5 には, 補正前の初期値をピンク軌跡で, 真値を水色で示す. 補正前の初期値は, 奥行き推定等の誤差により前後にバタついている. 軌跡補正 DNN の推定軌跡を見てみると, 実際の車両の動きのような滑らかな軌跡に補正できていることがわかる.

図 6 の歩行者の軌跡では, 各対象が横断歩道上を通過する様子が再現できている. 一方で, 細かく見てみる

表 2: 自車の位置・向き誤差. () は GPS 補正前の誤差.

	位置誤差 [m]	向き誤差 [deg]
ドラレコ GPS	2.5 (5.7)	-
Colmap [19]	1.7 (5.6)	5.5 (8.1)
同時最適化 [5]	3.5 (6.2)	8.3 (10.6)
DNN-Pose [9]	2.6 (5.9)	9.8 (9.3)
提案法	1.5 (5.6)	3.7 (6.2)

表 3: 対応点間引きの効果 (15 秒のシーンで比較).

	位置誤差 [m]	処理時間 [s]
Colmap [19]	1.0 (3.7)	1696
提案法 (間引きなし)	0.9 (3.7)	219
提案法 (間引きあり)	0.9 (3.7)	148
システム全体	-	305

と, 例えば Obj5 は, 実際よりも手前に近づくような軌跡になっているなどの乖離がある. 人が密集しているエリアでは, オクルージョンにより検出矩形内に対象以外の領域も含まれてしまい, その奥行き値に引きずられ実際の位置と違う場所にマッピングされてしまう. 数フレームのオクルージョンであれば, 軌跡補正 DNN で補正できるが, 定常的なオクルージョンの補正は難しく, 実際とは異なる形状の軌跡になってしまった.

6.5.2 定量評価

移動体の位置・向きの誤差を評価する. 比較手法には以下を用いる.

- Vanilla LSTM・・・軌跡補正 DNN を単純な LSTM に変更したモデル.
- 提案法 (Aug ランダム)・・・ネットワーク構造は提案法と同じだが, データ拡張では位置等を考慮せずにランダムで誤差を付与する.

評価指標には, 自転車同様に地図上の絶対位置と向きの誤差だけでなく, 各時刻の自転車を基準とした相対的な位置と向きの誤差を追加する. 自転車を基準とすることで, 自転車位置誤差の影響を排除する. また, 相対位置誤差には, L2 ノルムと Absolute Relative Error (AbsRel) の 2 つを導入する. AbsRel は, 推定値と真値の差の絶対値を, 真値で割ったものである. 遠方の対象ほど誤差が大きくなりやすいため, L2 ノルムの位置誤差は遠方の対象に引きずられやすい. 一方, AbsRel は真値位置における相対的な誤差率を表すので, 遠方の誤差に引きずられにくい特徴がある.

表 4 に, 対象・手法ごとの誤差を示す. 提案法が, 双方の対象で最も精度が高い. Vanilla-LSTM と提案法との誤差の差は, モデルを Bi-directional LSTM にした効果である. 入力軌跡の特徴として, 遠方になるほど誤差が大きくなる傾向があり, 単方向の LSTM は, 追い越し車両 (誤差小→大) は対応できるが, 近づく車両 (誤差大→小) は対応できなかった. 一方, Bi-directional LSTM は全時刻の情報を等しく考慮できるので, 両方の対象に対応でき, 誤差が小さくなった. 提案法と提案法 (Aug ランダム) の差からは, データ拡張における誤差モデルの効果を確認できる. 誤差モデルでは, 距離に応じた誤差量や, 遠方の対象は手前に推定されやすいといったバイアス成分の誤差も再現している. 一方, Aug ランダムは, 誤差平均は 0 で分散も一定値を与えている

表 4: 移動体の位置・向き誤差. () は GPS 補正前の誤差.

	絶対位置 [m]	絶対向き [deg]	相対位置 [m] (L2 ノルム)	相対位置 (AbsRel)	相対向き [deg]
車・バイク (評価数: 123+9)					
初期値	8.4 (10.7)	-	7.9	0.24	-
Vanilla LSTM	7.0 (9.2)	16.1 (19.2)	6.6	0.29	15.1
提案法 (Aug ランダム)	6.8 (8.9)	12.4 (15.2)	6.2	0.26	11.6
提案法	6.3 (8.7)	12.4 (15.9)	5.9	0.23	11.6
人・自転車 (評価数: 24+9)					
初期値	4.7 (6.9)	-	2.9	0.19	-
Vanilla LSTM	3.1 (7.1)	34.7 (33.9)	2.2	0.17	30.8
提案法 (Aug ランダム)	3.0 (7.1)	33.9 (33.6)	2.2	0.17	28.6
提案法	2.7 (6.8)	33.7 (33.3)	1.9	0.13	28.5

表 5: 道路附属物の位置誤差.

	位置誤差 [m]
軌跡補正 DNN	4.0 (5.4)
提案法	2.7 (5.0)

ため, 上記の前提知識は学習できていない. 誤差モデルをデータ拡張に導入することで, 実際の傾向に合わせた多様な軌跡を学習できたため, 精度向上に繋がった.

次に, 提案法におけるクラスによる誤差の差異に着目する. 位置誤差は人・自転車クラスの方が小さかった. これは人・自転車の評価可能な対象が, 自転車から近いものが多かったためである. 人や自転車は体積が小さく, 遠方の対象から LiDAR 点群を十分に得られず教示ができなかったことが起因している. 一方, 向きの誤差は, 人・自転車クラスの方が大きかった. これは, 人・自転車は移動速度が小さく, 向きの推定が難しかったり, 定性評価で言及したようにオクルージョンが発生しやすく軌跡形状が歪みやすかったりするためだと考える.

6.6 道路附属物位置推定における評価

図 1, 6 に道路附属物の位置推定結果を示す. 衛星写真から想定される真値位置の近くに推定できていることがわかる. 定量評価では, SfM 点群を用いる提案法と, 移動体の推定で用いた位置補正 DNN による結果を比較する. 表 5 に各手法における地図上の位置誤差を示す. 提案法の方が誤差は小さく, 手法の有効性を確認できる. 位置補正 DNN で誤差が大きい要因は 2 つ考えられる. 1 つ目は, 前段の奥行き推定 DNN の誤差が大きいことである. 信号機や標識などはサイズが小さく, 奥行き推定 DNN の学習が難しいため, 奥行き推定の誤差が大きくなり, 補正もうまくできなかった. 2 つ目は, 軌跡補正 DNN では停止物の軌跡推定が難しいことである. 誤差のある入力軌跡から, 対象が低速で動いているか停止しているかを判断することは難しい. これらの理由から, 停止物であることを前提とした SfM による最適化の方が高精度に推定できたと考える. 以上より, 移動体と道路附属物のような静止体で, 位置推定の方法を分ける提案法は有効であると言える.

7 おわりに

本論文では, 安価かつ手軽なドラレコで, 道路環境解析を行うために, 自転車や, 周辺車両等の移動体, 道路附属物等の静止体の位置・軌跡を推定する一連のアルゴリズムを提案した. 自転車の位置・軌跡推定では, GPS 誤差

が大きいドラレコデータの特性を考慮して, 3 つのステップからなる手法を提案することで, 安定かつ高精度な自転車位置推定を実現した. SLAM / SfM では原理的に算出できない移動体の位置は, DNN による奥行き推定と時系列補正を用いることで, 高精度かつ滑らかな軌跡の推定を実現した. 今後は, より広範なエリアでの検証や, 車載以外の用途への検証を行っていく.

参考文献

- [1] 国土交通省, “道路分野における新技術導入促進方針,” 2020.
- [2] 田中優, et al., “エッジデバイスデータの AI 解析によるドライバーサービスの高度化機能,” in 東芝レビュー, 2022.
- [3] 堤盛人, et al., “地方自治体における道路維持管理業務のための道路構造物に関する情報の利活用方策,” in 新道路技術会議, 2021.
- [4] Hartley, R., et al., “Multiple view geometry in computer vision,” in Cambridge university press, 2003.
- [5] Chen, X., et al., “Integration of low-cost gnss and monocular cameras for simultaneous localization and mapping,” in Sensors, 2018.
- [6] Schreiber, M., et al., “Vehicle localization with tightly coupled GNSS and visual odometry,” in IV, 2016.
- [7] Wu, H., et al., “Virtual Sparse Convolution for Multimodal 3D Object Detection,” in CVPR, 2023.
- [8] Piccinelli, L., et al., “iDisc: Internal Discretization for Monocular Depth Estimation,” in CVPR, 2023.
- [9] Zhou, T., et al., “Unsupervised learning of depth and ego-motion from video,” in CVPR, 2017.
- [10] Ranftl, R., et al., “Vision Transformers for Dense Prediction,” in ICCV, 2021.
- [11] Googl Earth, <https://www.google.co.jp/earth/>.
- [12] Tian, Z., et al., “FCOS: Fully Convolutional One-Stage Object Detection,” in ICCV, 2019.
- [13] Poudel, R. P., et al., “Fast-scnn: Fast semantic segmentation network,” in BMVC, 2019.
- [14] Alcantarilla, P. F., et al., “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” in IEEE Trans. Patt. Anal., 2011.
- [15] Geiger, A., et al., “Vision meets robotics: The kitti dataset,” in IJPR, 2013.
- [16] Horn, B. K., “Closed-form solution of absolute orientation using unit quaternions,” in Josa a, 1987.
- [17] Kawasaki, A., et al., “Multimodal trajectory predictions for urban environments using geometric relationships between a vehicle and lanes,” in ICRA, 2020.
- [18] Schuster, M., “Bidirectional recurrent neural networks,” in IEEE Trans on Signal Processing, 1997.
- [19] Schonberger, J. L., et al., “Structure-from-motion revisited,” In CVPR, 2016.

本稿に記載されている社名・商品名・サービス名などは, それぞれ各社が商標として使用している場合がある.