# MUTR3D: 3D-2Dクエリによるマルチカメラトラッキングフレームワーク

Tianyuan Zhang
Carnegie Mellon University
tianyuaz@andrew.cmu.edu

Xuanyao Chen
Fudan University
xuanyaochen19@fudan.edu.cn

Yue Wang
マサチューセッツ工科大学
yuewang@csail.mit.edu

Yilun Wang
Li Auto
yilunw@cs.stanford.edu

Hang Zhao ¶
Tsinghua University
hangzhao@mail.tsinghua.edu.cn

## Abstract

複数のカメラからの正確で一貫性のある3Dトラッキングは、ビジョンベースの自律走行システムにおいて重要な要素である。複数のカメラにまたがる複雑なシーンにおける3D動的オブジェクトのモデリングを含む。この問題は、奥行き推定、視覚的オクルージョン、外観の曖昧さなどにより、本質的に困難である。さらに、オブジェクトは時間やカメラによって一貫して関連付けられていない。そこで、MUTR3Dと呼ばれるエンドツーエンドのMUlticamera TRackingフレームワークを提案する。先行研究とは対照的に、MUTR3Dはオブジェクトの空間的・外観的類似性に明示的に依存しない。その代わりに、我々の手法は、複数のカメラと複数のフレームに現れる各オブジェクトの空間的および外観的なコヒーレントトラックをモデル化するために、3Dトラッククエリを導入する。カメラ変換を用いて、3Dトラッカーとその2D画像における観察結果をリンクさせる。各トラッカーは、カメラ画像から得られる特徴に従ってさらに改良される。MUTR3Dは、予測されたトラッキング結果とグランドトゥルースとの差を測定するために、セット間損失を使用する。そのため、非最大抑制や/バウンディングボックスの関連付けなどの後処理を必要としない。MUTR3DはnuScenesデータセットにおいて、5.3 AMOTAで最先端手法を上回った。コードはhttps://github.com/a1600012888/MUTR3Dで入手可能。

## 1. Introduction

3Dトラッキングは、自律走行、ロボット工学、バーチャルリアリティなど、様々な知覚システムにおいて極めて重要である。最も基本的なインカーネーションでは、3Dトラッキングはフレームごとのオブジェクトを予測し、それらの間の対応関係を時間的に見つけることを含む。フレームごとの物体検出結果が与えられた場合、この問題は、オブジェクトの類似性に応じて、フレームをまたいでオブジェクトを首尾一貫した方法で関連付けることに帰着する。一方、トラッキングは検出の安定性を向上させ、フレーム間の検出予測の一貫性を強制する。しかし、これは複雑な反復最適化問題を誘発する。

マルチカメラのケースを詳述する場合、より多くの課題が生じる。まず、正確なトラッキングのためには、正確な3D検出が必要である。しかし、カメラによる3次元物体検出は未解決の問題である。第二に、視覚トラッカーは、複雑なシーンにおけるオクルージョンや外観の曖昧さに関して脆弱である。例えば、興味のある人が車の後ろを歩き、数秒後に別のポーズで再登場することがある。第三に、トラッカーはカメラビューの境界を越えて移動するオブジェクトを失うことが多い。したがって、時間的な関連付けだけでなく、空間的に一貫した予測を行うために、物体が異なるカメラにまたがったり、交差したりするときにも、カメラ間の関連付けを行う必要がある。これらの課題は、3Dビジョントラッカーの実用化を妨げている。

視覚に基づく3Dオブジェクト追跡に関する研究はほんの一握りである。古典的なカルマンフィルタリングに基づく手法[38]は、任意の検出器からの検出結果を入力とし、さらに時間経過に伴う物体の状態推定と関連付けを行う。より最近の学習ベースの手法も、検出から追跡へのパラダイムに従っており、まず各フレームに対してオブジェクト提案を行い、次にそれらを特徴空間においてディープニューラルネットワークと関連付ける[8, 11, 40]。

本研究では、空間的類似性と外観的類似性を用いて、

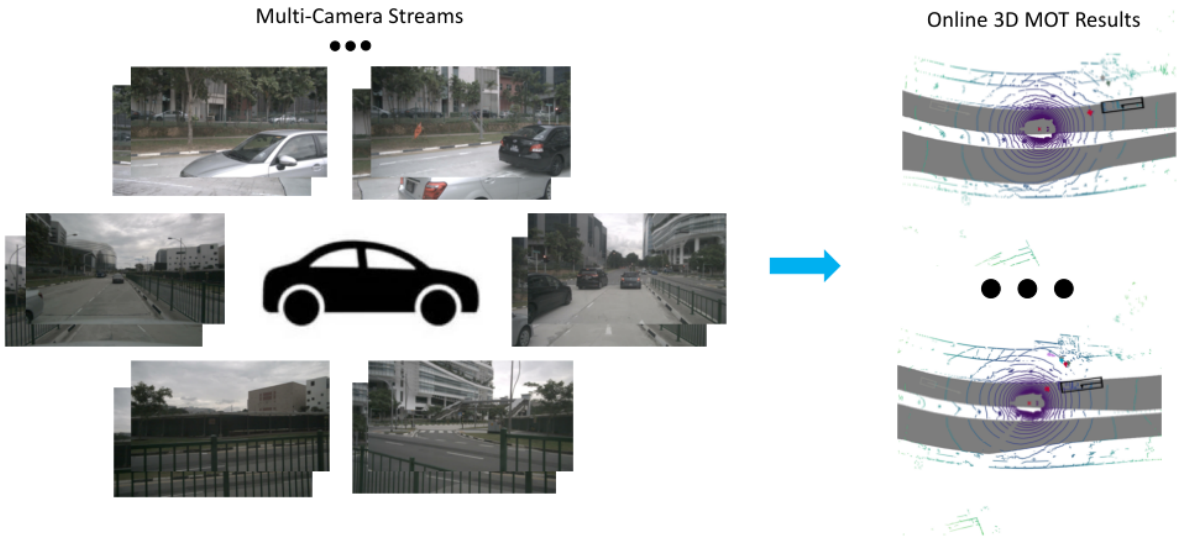


図1. MUTR3Dと名付けられたエンドツーエンドのマルチカメラ3Dトラッキングフレームワークを提案する。我々のアルゴリズムは、既知のパラメータを持つ任意のリグで動作する。マルチカメラ3D検出と、カメラ横断的なクロスフレームオブジェクトアソシエーションをエンドツーエンドで処理する。

オブジェクトを3Dトラックに関連付けるオンラインマルチカメラ3DマルチオブジェクトトラッキングフレームワークであるMUTR3Dを提案する。より具体的には、オブジェクトトラックの3D状態と外観の特徴を、時間経過やカメラ間で直接モデル化する3Dトラッククエリを紹介する。各フレームで、3Dトラックは、すべての可視カメラから特徴をサンプリングし、トラックの作成/追跡/終了を学習する。MUTR3Dは、これまでの研究とは対照的に、統一されたエンドツーエンドのフレームワークで、検出と追跡を同時に行う。フレーム間で同じクエリからデコードされたオブジェクトは、本質的に関連付けられる。

要約すると、我々の貢献は3つある：

- 我々の知る限り、MUTR3Dは初の完全なエンドツーエンドのマルチカメラ3Dトラッキングフレームワークである。明示的なトラッキングヒューリスティックを使用する既存の検出-追跡手法とは異なり、我々の手法はオブジェクトトラックの位置と外観の分散を暗黙的にモデル化する。さらに、非最大抑制、バウンディングボックスの関連付け、オブジェクトの再識別(Re-ID)など、一般的に使用される後処理ステップを排除することで、3Dトラッキングパイプラインを簡素化する。
- 物体のトラック全体の3D状態をモデル化する3Dトラッククエリを紹介する。3Dトラッククエリサンプルは、すべての可視カメラから特徴を抽出し、トラックをフレームごとにエンドツーエンドで更新する。
- 我々のエンドツーエンドの3Dトラッキング手法は、NuScenesビジョンのみの3Dトラッキングデータセットにおいて、27.0%のAMOTAで最先端の性能を達成した。より具体的には、MUTR3Dはマルチカメラ設定において、IDスイッチを12%削減し、従来のSOTA手法よりもはるかに優れた性能を発揮する。
- 現在の3Dトラッカーでモーションモデルを評価するために、2つのメトリクスを提案する：平均トラッキング速度誤差(ATVE)とトラッキング速度誤差(TVE)である。追跡されたオブジェクトの推定モーションの誤差を測定する。

## 2. Related Work

### 2.1. 自律走行における3D MOT

自律走行車の場合、周囲の物体の位置、向き、大きさ、速度を推定しながら追跡することが重要である。最近の3D検出の進歩[14, 28, 41, 43, 49]により、最新の3D MOTはトラッキング-バイ-ディテクションパラダイムに従っている。これらの方法は、現在のフレーム内のオブジェクトを検出し、それらを以前のトラックレットと関連付ける。Wengら[38]は、シンプルかつ効果的な関連付け方法をベンチマークしている。カルマンフィルタリングにより以前のトラックレットの位置を予測し、3D IoUを用いて現在の検出を関連付ける。IoU以外にも、L2距離[43]や一般化3D IoU[21]を用いて、3Dボックスと純粋な位置手がかりを関連付けた研究がいくつかある。多くの研究は、学習された動きや外観の特徴を追加したり[2, 8, 9]、グラフニューラルネットワークを使用したり[5, 39, 45]することで、より高度な関連付けを使用している。いくつかの研究は、検出スコアからの手がかりを利用することで、ライフサイクル管理を改善する方法を研究している[3,21]。QD3DT 現在のSOTA(State-of-The-Art)カメラベースのトラッキングアルゴリズムは、密なコントラスト学習によって外観マッチング特徴を学習する。

最後に、視覚的特徴、モーションキュー、深度順序を組み合わせて関連付けを行う。RGB外観の手がかりは強いが、カメラベースの3D MOT [2, 8, 11, 27, 32, 47]の性能は、LiDARベースに比べて遅れている。nuScenes 3D MOTチャレンジのパブリックリーダーボードでは、STOAカメラベースの手法は21.7%のAMOTAを達成し、STOA LiDARベースの手法は67.9%のAMOTAを達成した。複数の異なる視点からの追跡の問題も注目されている[30]。

### 2.2. カメラベースの3D検出

近年、3次元物体検出は大きな進歩を遂げている。2次元検出フレームワーク[28, 34, 42, 48]をベースにしたアルゴリズムの流れ。インスタンスの深さとスケールの基本的な曖昧さを解決するために、カテゴリカルな正準形状[1,19]、幾何学的関係グラフ[33]、事前学習された単眼深度[18,22]が使用される。もう一つの手法は、3次元空間やBirds-EyeView上の表現を扱うものである。Pseudolidar [35, 44]は、事前に訓練された単眼深度モデルを使用して、ピクセルを3D点群に持ち上げ、LiDARベースの検出器を使用して3D検出を実行する。Lift-Splat-Shot[23]は、リフティングプロセスを完全に微分可能にし、リフティングモジュールを下流のタスクと共同で学習させる。その後、CaDDN [24]とBEVDet [12]は、3D検出のために同様の表現を使用した。DETR3D [36]は逆投影プロセスを採用し、クエリベースのマルチカメラ3D検出器を構築する。遠近法画像平面を直接扱う場合と比較して、3D空間で扱う場合の大きな利点は、任意のカメラリグを採用し、複数のセンサ特徴を融合することが容易であることである。現在のところ、性能に明確な利点はない[22]。より多くの比較はまだ十分に検討されていない。

## 2.3. クエリベースの検出と追跡

現代の検出と追跡のアプローチの支配的なタイプは、検出のタスクをピクセル単位の回帰と分類に減らし[13, 17, 25, 26, 31, 48]、次に検出ボックスを関連付けて追跡を実行することである。最近、DETR [7]は、クエリベースの集合予測を用いて、最先端の検出結果を達成することに成功した。その後のTrackFormer [20]、MOTR、TransTrack [29, 46]はこのアイデアをオンライン2D MOTに拡張した。我々の研究は、クエリベースのトラッキングのフレームワークに基づいている。本フレームワークをモーションモデルを用いたマルチカメラ3D MOTに拡張する。

# 3. Methods

## 3.1. クエリに基づく物体追跡

我々のアルゴリズムにはクエリベースのトラッキングを採用する。クエリベースのトラッキングは、クエリベースの検出[7]から拡張され、固定サイズの埋め込みセットである検出クエリが、2Dオブジェクト候補を表現するために使用される。トラッククエリは、検出クエリの概念をマルチフレームに拡張する、すなわち、フレームをまたいでトラックレット全体を表現する[20,37,46]。具体的には、各フレームの最初に新生クエリのセットを初期化し、次にクエリをフレームごとに自動回帰的に更新する。デコーダヘッドは、各フレームの各トラッククエリから1つのオブジェクト候補を予測し、同じトラッククエリから異なるフレームでデコードされたボックスは直接関連付けられる。適切なクエリライフサイクル管理により、クエリベースのトラッキングはオンライン方式で共同検出と追跡を行うことができる。

クエリベースのマルチカメラ3Dトラッカーには、3つの重要な要素がある。(1) クエリベースのオブジェクト追跡損失は、2つの異なるタイプのクエリ、新生クエリ、古いクエリに対して異なる回帰ターゲットを割り当てる。(2) マルチカメラスパースアテンションは、各クエリの画像特徴をサンプリングするために3次元参照点を使用する。(3) モーションモデルはオブジェクトのダイナミクスを推定し、フレーム間でクエリの参照点を更新する。図2に我々のトラッカーの流れを示す。

### 3.2. エンドツーエンドの物体追跡損失

まず、クエリベーストラッキングの文脈で、ラベル割り当ての概念を説明する。我々のアルゴリズムは、フレーム間で変化するトラッククエリのセットを維持する。現在のフレームでは、各クエリから1つのオブジェクト候補をデコードする。理想的には、同じクエリからデコードされたオブジェクト候補は、フレーム間で同じオブジェクトを表す必要があり、その結果、トラックレット全体が形成される。クエリベースのトラッカーを訓練するために、各フレームの各クエリに対して1つのターゲットグランドトゥルースオブジェクトを割り当てる必要があり、割り当てられたグランドトゥルースオブジェクトはクエリの回帰ターゲットとして機能する。具体的には、ラベル割り当ては、グランドトゥルースオブジェクトとトラッククエリ間のマッピング関数である。我々は通常、マッピングが1対1のマッピングであることを保証するために、$\varphi$(オブジェクトなし)を持つグランドトゥルースオブジェクトの集合を予測オブジェクト候補の数にパッドする。現在のフレームでN個のデコードされたオブジェクト候補{$\hat{y}_1$,.. $\hat{y}_N$ }があるとすると、ラベル割り当ては写像$\pi$ ∈ {1, 2.. N } 7→ {1, 2.., N }.そして、学習損失はペアボックス損失の和として表現できる:

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{\text{box}}(y_{\pi(i)}, \hat{y}_i), \tag{1}$$

ここで、$y_{\pi(i)}$は割り当てられた目標地上真理オブジェクトを表し、$L_{box}$は任意のバウンディングボックス損失である。各フレームには2種類のクエリがあり、それぞれ異なるラベル割り当て戦略を持つ。新生児のクエリは学習されたクエリの集合である。これらは入力にとらわれず、各フレームの最初にクエリのセットに追加される。新生児のquiresは、現在のフレームで新しく出現した物体を検出する責任がある。そこで、新しく出現したグランドトゥルースオブジェクトと新生クエリからのオブジェクト候補をDETR [7]として二分割マッチングする。

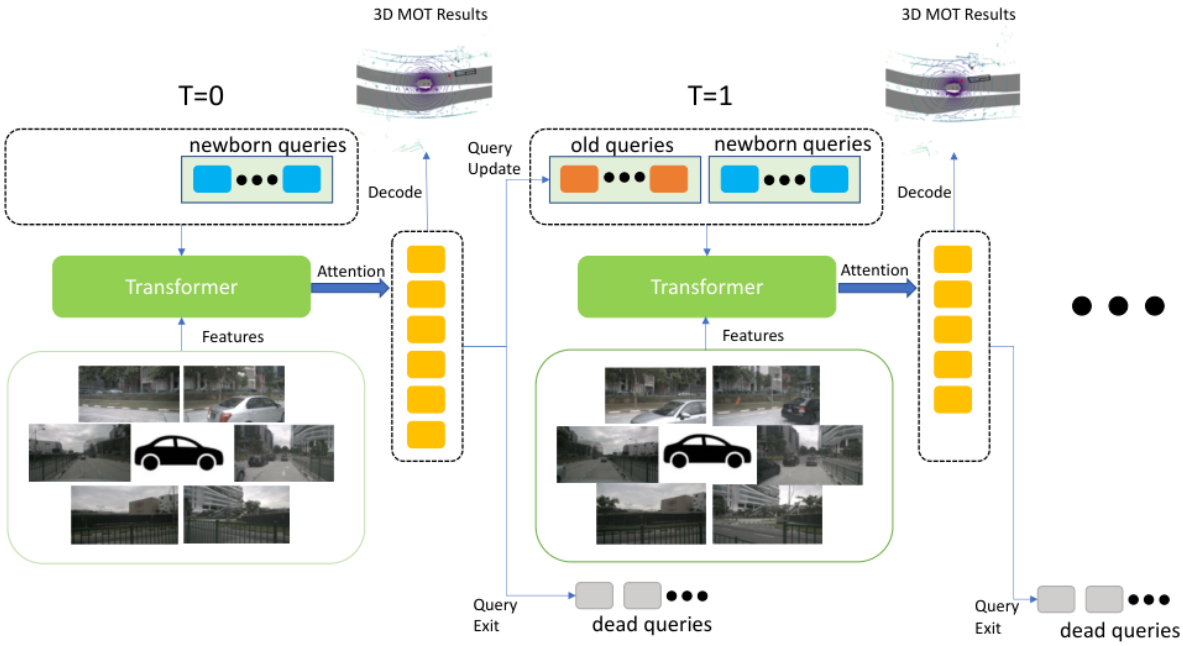


図2. オンライン・マルチカメラ・トラッカーのパイプライン概要。黒破線枠内の小さな色の四角はすべてトラッククエリを表す。青いボックスは新生クエリを表し、各フレームの最初にトラッククエリのセットに追加される学習可能なクエリの固定セットである。オレンジ色のボックスは古いクエリを表し、前のフレームからのアクティブなクエリである。トラッククエリは、現在のフレームでオブジェクト候補をデコードするために、マルチカメラ特徴に注目する。次に、非アクティブなクエリをフィルタリングする。また、オブジェクトの動きやエゴの動きを補正するために、アクティブなクエリの参照点を更新する。最後に、更新されたクエリは、同じオブジェクトを追跡するために、次のフレームに渡された。

古いクエリは、オブジェクトの検出や追跡に成功した前のフレームからのアクティブなクエリである。古いクエリは、現在のフレームで以前に出現したオブジェクトを追跡する役割を担っている。古いクエリに対する割り当ては、グランドトゥルースオブジェクトを検出することに初めて成功した後に固定される。同じオブジェクトが現在のフレームにある場合は追跡するように割り当てられ、そうでない場合は$\varphi$(オブジェクトなし)。

式1の3次元ボックス損失$L_{box}$は次のように定義される:

$$\mathcal{L}_{\text{box}}(y_{\pi(i)}, \hat{y}_j) = \begin{cases} L_{\text{cls}}(c_{\pi(i)}, \hat{c}_j) + \lambda L_{\text{reg}}(b_{\pi(i)}, \hat{b}_j) & y_{\pi(i)} \neq \varnothing \\ L_{\text{cls}}(c_{\pi(i)}, \hat{c}_j) & y_{\pi(i)} = \varnothing \end{cases} \tag{2}$$

$L_1$損失を$L_{reg}$に用い、$L_{cls}$を焦点損失[17]とし、3次元物体$y_{\pi(i)}$はクラスラベル$c_{\pi(i)}$とバウンディングボックスパラメータ$b_{\pi(i)}$を用いてパラメータ化し、パラメータ化の詳細は式6に示す。

### 3.3. マルチカメラトラッククエリデコーディング

我々の変換デコーダヘッドは、トラッククエリを受け取り、マルチカメラ画像特徴量とそれらに注目し、抽出されたクエリ特徴量はオブジェクト候補のデコードに使用される。我々のデコーダは、クエリ間の自己注意と、クエリと画像特徴間の交差注意の2種類の注意モジュールを持つ。

メモリ効率のために、DETR3D [36]の参照点ベースの注意を採用し、交差注意を行う。本節の表記では、3次元座標またはその2次元投影のみを太字で表記する。例えば、基準点の3次元座標$\mathbf{c}_i$ 、推定速度$\mathbf{v}_i$ 。

クエリの初期化。各クエリが初期化されるとき、つまりあるフレームで新生クエリとして導入されるときに、3D参照点$\mathbf{c}_i$を割り当てる。3次元参照点は、共有MLP(多層パーセプトロン)を用いて、学習可能な埋め込みからデコードされる:

$$\boldsymbol{c}_i = \Phi^{\text{ref}}(q_i), \tag{3}$$

ここで、$q_i$は学習可能なクエリ埋め込みを表し、3D参照点は変換デコーダの層を通して、フレームをまたいで自動回帰的に更新される。物体候補の3次元位置を近似することを目的とする。

クエリ特徴抽出。クロスアテンションは、各クエリの参照点をすべてのカメラに投影し、点の特徴をサンプリングすることで機能する。各フレームについて、M台のカメラから画像を同期させたとする。各画像のピラミッド型特徴量を独立に抽出する。

ピラミッド型特徴量の集合を次のように表す：$f_1$，$f_2$，$f_3$，$f_4$.各項目$F_k = \{F_{k1}, .. F_{kM}\}$，$F_{ki} \in R^{H \times W \times C}$ はM個の画像の特徴量に対応する。提供されるカメラ投影行列を $T = \{T_1, .. T_M\}$，$T_i \in R^{3 \times 4}$.具体的には、サンプリングされた点特徴量$f_{ci}$は：

$$\begin{aligned} \boldsymbol{c}_{mi} &= T_m(\boldsymbol{c}_i \oplus 1), w_i = \mathrm{MLP}(q_i), \\ f_{\boldsymbol{c}_i} &= \sum_{k}^{4} \sum_{m}^{M} F_{km}(\boldsymbol{c}_{mi}) \cdot \sigma(w_{kmi}), \end{aligned} \tag{4}$$

ここで、$c_{mi}$はカメラmの画像平面上に投影された2次元座標を表し、$F_{km}$（$c_{mi}$）は画像特徴からのバイリニアサンプリングを表し、σ()はシグモイド関数を表し、重み付け係数を正規化するために用いられる。
次に、抽出された特徴量を用いて、クエリとその参照点を更新する

$$\begin{aligned} q_i &\leftarrow q_i + \mathrm{MLP}(f_{\boldsymbol{c}_i} + \mathrm{PE}(q_i)), \\ \boldsymbol{c}_i &\leftarrow \boldsymbol{c}_i + \mathrm{MLP}(f_{\boldsymbol{c}_i}), \end{aligned} \tag{5}$$

ここで、PEは学習可能な位置エンコーディングであり、各クエリで初期化される。トランスフォーマーデコーダーの層の後、最終的なクエリ特徴を使って、現在のフレームのオブジェクト候補をデコードする。

3次元物体パラメトリゼーション。2つの小さなFFNを用いて、3Dボックスパラメータとカテゴリラベルをデコードする。3次元ボックスをさらに10次元のパラメータでパラメータ化する：エゴフレームにおけるボックス中心の座標 $x_i \in R^3$，3次元ボックスのサイズ $s_i = (w_i, l_i, h_i) \in R^3$，エゴフレームにおける2次元速度 $v_i = (v_i^x, v_i^y) \in R^2$ と方位（$\sin \theta_i$，$\cos \theta_i$），ここで$\theta_i$はエゴフレームにおけるヨー角である。箱の中心の座標は、基準点に残差を加えることで予測されます：

$$\boldsymbol{x}_i = \boldsymbol{c}_i + \mathrm{MLP}(q_i). \tag{6}$$

### 3.4. クエリ寿命管理

オンライン方式で消滅するオブジェクトに対処するには、各フレームの後に非アクティブなクエリを削除する必要がある。各クエリの信頼度スコアを、予測されたボックスの分類スコアと定義する。ボックススコアには2つの閾値パラメータ$\tau_{new}$と$\tau_{old}$を用い、ライフマネジメントを制御するために時間長Tを用いる。
推論中、各フレームの新生クエリについて、スコアが$\tau_{new}$より低い場合、それを除去する。古いクエリの場合、連続するTフレームでスコアが$\tau_{old}$より低かった場合は削除する。nuScenesデータセットでは、$\tau_{new} = 0.4$、$\tau_{old} = 0.35$、$T = 5$を数回のトレイル後に選択する。
学習中、$\varphi$にマッチしたクエリは非アクティブとみなす。現在のフレームの新生クエリに対して、$\varphi$,にマッチする場合は削除する。古いクエリについては、連続するT回$\varphi$にマッチした場合は削除する。マッチングされたが削除されなかった古いクエリは、変換デコーダを通して更新され続けることに注意してください。

### 3.5. クエリ更新とモーションモデル

古い(死んだ)クエリをフィルタリングした後、トラッククエリ、その特徴と3D参照点の両方を更新する。3次元基準点を更新する目的は、物体のダイナミクスをモデル化し、自我運動を補正することである。3Dトラッキングでよく使われる2つのモーションモデル、例えばカルマンフィルタ[21, 38]は、未知の速度を推定するためにフレーム間の観測位置を使用し、例えばCenterTrack[43, 47]は検出器から予測された速度を使用する。クエリから予測される速度を利用し、フレームを経由して更新し、マルチフレーム特徴を集約することができる。エゴフレーム速度を予測するために、小さなFFNを使用する。予測された速度はグランドトゥルースで教師される。現在のフレームと次のフレームのエゴポーズを $R_t$，$R_{t+1} \in R^{3\times3}$，$T_t$，$T_{t+1} \in R^3$ とする。この2つのフレーム間の時間差を△tとする。予測ボックス速度 $v_i = (v_i^x, v_i^y, 0) \in R^3$ を用いて、i番目のクエリの参照点 $c_i$ を更新する：

$$\boldsymbol{c}_i \leftarrow R_{t+1}^{-1}(R_t(\boldsymbol{c}_i + \boldsymbol{v}_i \times \Delta t) + T_t - T_{t+1}). \tag{7}$$

複数フレームの外観変化を暗黙的にモデル化するために、前のフレームの特徴を用いてトラッククエリを更新する。MOTR[46]に従い、メモリバンクと名付けられたアクティブなクエリのそれぞれについて、固定サイズの先入れ先出しキューを維持する。各フレームの後、各クエリとそのメモリバンクに対してアテンションモジュールを適用する。トラッククエリは注意モジュールのクエリとして機能し、対応するメモリバンクはキーと値のセットとして機能する。

## 4. Experiments

### 4.1. Datasets

全ての実験に nuScenes [6] データセットを使用する。実世界1000シーケンス、トレーニング用700シーケンス、検証用150シーケンス、テスト用150シーケンスから構成される。各シーケンスにはおよそ40の注釈付きキーフレームがある。キーフレームは各センサーの同期フレームで、サンプリングレートは2FPSである。各フレームには、360度の全視野を持つ6台のカメラからの画像が含まれる。7つのオブジェクトカテゴリに対する3Dトラッキングアノテーションを提供する。

### 4.2. 評価指標

平均多オブジェクト追跡精度(AMOTA)と平均多オブジェクト追跡精度(AMOTP)は、nuScenes 3Dトラッキングベンチマークの主要な指標である。AMOTAとAMOTPは積分によって計算される。

表1. nuScenesデータセットにおける最先端手法との比較。公開カメラベースの3Dトラッキングにおいて、我々のアルゴリズムは最先端の結果を達成し、検証セットではAMOTAで0.052、テスト分割では0.053でQD3DT [11]を上回った。

| | Modality | AMOTA ↑ | AMOTP ↓ | RECALL ↑ | MOTA ↑ | IDS ↓ | #params |
|---|---|---|---|---|---|---|---|
| Validation Split | | | | | | | |
| CenterPoint [43] | LiDAR | 0.665 | 0.567 | 69.9% | 0.562 | 562 | 9M |
| SimpleTrack [21] | LiDAR | 0.687 | 0.573 | 72.5% | 0.592 | 519 | 9M |
| DEFT [8] | Camera | 0.201 | N/A | N/A | 0.171 | N/A | 22M |
| QD3DT [11] | Camera | 0.242 | 1.518 | 39.9% | 0.218 | 5646 | 91M |
| Ours | Camera | **0.294** | **1.498** | **42.7%** | **0.267** | **3822** | 56M |
| Test Split | | | | | | | |
| CenterTrack [47] | Camera | 0.046 | 1.543 | 23.3% | 0.043 | **3807** | 20M |
| DEFT [8] | Camera | 0.177 | 1.564 | 33.8% | 0.156 | 6901 | 22M |
| QD3DT [11] | Camera | 0.217 | 1.550 | 37.5% | 0.198 | 6856 | 91M |
| Ours | Camera | **0.270** | 1.494 | **41.1%** | **0.245** | 6018 | 56M |

MOTA(多オブジェクト追跡精度)とMOTP(多オブジェクト追跡精度)の値を全リコールにわたって示す:

$$\text{AMOTA} = \frac{1}{L} \sum_{r \in \{\frac{1}{L}, \frac{2}{L}, \ldots, 1\}} \text{MOTA}_r, \tag{8}$$

$$\text{MOTA}_r = \max\left(0, 1 - \frac{\text{FP}_r + \text{FN}_r + \text{IDS}_r - (1-r)\text{GT}}{r\text{GT}}\right), \tag{9}$$

ここで、$\text{FP}_r$ 、$\text{FN}_r$ 、$\text{IDS}_r$ は、対応するリコールrで計算された偽陽性、偽陰性、アイデンティティスイッチの数を表す。GTはグランドトゥルースのバウンディングボックスの数である。AMOTAは次のように定式化できる:

$$\text{AMOTP} = \frac{1}{L} \sum_{r \in \{\frac{1}{L}, \frac{2}{L}, \ldots, 1\}} \frac{\sum_{i,t} d_{i,t}}{\text{TP}_r}, \tag{10}$$

ここで、$d_{i,t}$は時刻tにおけるマッチしたトラックiの2次元鳥瞰位置誤差を表し、$\text{TP}_r$は対応するリコールrで計算されたマッチの数を示す。
また、CLEAR [4]やLiら[15]のMOTA、MOTP、IDSなどのトラッキングメトリクスも報告する。これらのメトリクスの信頼度閾値は、各カテゴリーについて最も高いMOTAを持つ閾値を独立に選択することによって選択される。

## 4.3. 実装の詳細

画像特徴抽出器には、先行研究[34] [36]に従い、変形可能な畳み込みを用いたResNet-101 [10]とFPN [16]を用いる。アブレーション研究のために、メモリ効率のためにResNet-101をResNet-50に置き換えた。

トレーニングの詳細 DETR3D[36]の3D検出事前トレーニングモデルを使用する。次に、頭部を置き換えて、3フレームのビデオクリップで72エポックのトラッカーを訓練する。

カルマンフィルタのベースラインカルマンフィルタベースの手法は、データセットを横断するLiDARベースの3Dトラッキングにおいて、最先端のトラッカーとなっている[21]。しかし、カメラベースのSOTA手法は、通常、学習された外観と動きの特徴をマッチングに利用する。カメラベース3D MOTの分野をさらに理解するために、DETR3D[36]検出器を用いた2つのカルマンフィルタベースラインを提供する。(1) 高度な設計を行わない基本バージョン。基本バージョンはAB3DMOTの公開実装[38]を改良したものである。低フレームレートデータとの関連付け時のIoU(Intersection over Union)の失敗を処理するために、3D IoUを計算する際に予測ボックスを20%拡大する。(2) また、3次元一般化IoUと2段の関連付けを用いたSimpleTrack [21]のカルマンフィルタベースラインの先進版も提供する。SimpleTrackはLiDARベースのMOTでSOTAの結果を得た。

## 4.4. 最先端技術との比較

表1に我々の手法とSOTA手法の比較を示す。カメラベースのトラッカーでは、現在のSOTA手法を大きく上回る。現在のSOTA手法QD3DT [11]によるAMOTAの利得は、検証セットで5.2ポイント以上、テストセットで5.3ポイント以上である。我々のトラッカーは、QD3DT [11]のように、NMSもアソシエーションステージもなく、エンドツーエンドで動作する。

カルマンフィルタの2つのベースラインの比較を表2に示す。我々はカルマンフィルタの基本バージョンを上回る。しかし、SimpleTrack [21]のよりテーラーメードのベースラインと比較すると、AMOTA、MOTA、MOTPのようなメトリクスではわずかな改善しか見られない。

## 4.5. モーションモデルの評価

動きモデルは、3D多オブジェクト追跡のための主要な手がかりの1つを提供する。運動モデルは、トラックレットの移動パターンを記述することを目的とする。異なるトラッキングアルゴリズムのモーションモデルを評価するために、

表2. nuScenes検証分割におけるカルマンフィルターベースの手法との比較。事前学習した検出器DETR3D[36]を用いて、2つのカルマンフィルタベースラインを構築する。アウトトラッカーと比較する。

| | AMOTA ↑ | AMOTP ↓ | RECALL ↑ | MOTA ↑ | MOTP ↓ | IDS ↓ |
|---|---|---|---|---|---|---|
| DETR3D [36] + KF | 0.263 | 1.569 | 39.7% | 0.260 | 0.952 | 4698 |
| DETR3D + SimpleTrack [21] | 0.293 | **1.307** | 41.8% | 0.263 | 0.84 | **1695** |
| Ours | **0.294** | 1.498 | **42.7%** | **0.267** | **0.799** | 3822 |

表3. 速度推定を評価する。nuScenes検証スプリットにおけるATVE(平均追跡速度誤差)とTVE(追跡速度誤差)を報告する。カルマンフィルタを用いた運動モデルと比較して、本手法はより良いTVEを得ることができる。

| | Modality | ATVE ↓ | TVE ↓ |
|---|---|---|---|
| CenterPoint [43] | LiDAR | 0.572 | 0.298 |
| QD3DT [11] | Camera | 1.876 | 1.373 |
| DETR3D + SimpleTrack | Camera | **1.344** | 0.836 |
| Ours | Camera | 1.548 | **0.768** |

AMOTPとMOTPの考え方に従い、平均トラッキング速度誤差(ATVE)とトラッキング速度誤差(TVE)の2つの指標を開発した。ATVEは次のように計算できる:

$$\text{ATVE} = \frac{1}{L} \sum_{r \in \{\frac{1}{L}, \frac{2}{L}, \ldots, 1\}} \frac{\sum_{i,t} ||v_i - v_t||_2}{\text{TP}_r}, \qquad (11)$$

ここで、一致した追跡予測値とグランドトゥルースの全てのペアをトラバースし、予測速度$v_i$とグランドトゥルース速度$v_t$の間の$L_2$誤差を計算する。平均追跡速度誤差は、すべてのリコールrを平均して計算され、$TP_r$は対応するリコールrのマッチ数を表す。MOTPと同様に、Tracking Velocity Errorは、最も高いMOTAを持つリコールで計算された平均速度誤差である。表3にモーションモデルの評価を示す。従来の最先端カメラトラッカーQD3DT[11]と比較して、我々の速度はより正確である。カルマンフィルタリングに基づく運動モデルと比較して、本アルゴリズムはより良いトラッキング速度誤差を達成する。

表4. 運動モデルに対するアブレーション動きモデルを削除すると、我々のアルゴリズムの性能は全てのメトリクスで低下する。

| | AMOTA | AMOTP | RECALL | MOTA | IDS |
|---|---|---|---|---|---|
| w/o Motion | 0.215 | 1.598 | 35.8% | 0.198 | 4100 |
| w/ Motion | **0.234** | **1.585** | **38.7%** | **0.22** | **3775** |

4.6. アブレーション研究

アブレーション研究では2つの要因を検討した。まず、モーションモデルを削除した場合の効果、すなわち、各フレームの終了時に3次元参照点を更新しない場合の効果を検討する。アブレーションの結果を表4に示す。モーションモデルを削除すると、全てのメトリクスで性能が低下する。

次に、学習フレーム数の効果を調べる。我々の手法は自動回帰的にオブジェクトを追跡し、教師強制は適用しない。学習中、後者のフレームで計算された勾配は、前のフレームでグラフを計算するために伝搬する。アブレーション研究では、ResNet-50バックボーンを用いて全ての実験を行う。表5に3,4,5フレームを用いた学習の性能を示す。その結果、学習フレーム数を増やすことで、徐々に性能が向上することがわかった。

表5. 学習フレーム数に対するアブレーション。より長いビデオクリップでモデルをトレーニングすることは有益である。

| #frames | AMOTA | AMOTP | RECALL | IDS | ATVE |
|---|---|---|---|---|---|
| 3 | 0.234 | 1.585 | 38.7% | **3775** | 1.606 |
| 4 | 0.242 | 1.580 | 39.7% | 4623 | 1.545 |
| 5 | **0.251** | **1.573** | **39.9%** | 3873 | **1.565** |

### 4.7. Qualitative results

図3は、8秒間のクリップのBEVとカメラビューの両方におけるトラッキングアルゴリズムの視覚化である。車の左右にあるニアファイルオブジェクトは、通常、複数のカメラで切り捨てられており、これはマルチカメラ3Dトラッキングにとって大きな課題である。灰色と黒色の車はFront-LeftカメラとBackLeftカメラ(3-rd/7-thと4-th/8-thの行)で切り捨てられ、我々のアルゴリズムは正しく処理することが分かる。

## 5. Conclusion

エンドツーエンドのマルチカメラ3D MOTフレームワークを設計する。我々のフレームワークは、3D検出、エゴモーションとオブジェクトモーションの補正、カメラとフレームをまたいだオブジェクトの関連付けをエンドツーエンドで行うことができる。nuScenesテストデータセットにおいて、我々のトラッカーは、現在の最先端のカメラベースの3DトラッカーQD3DT [11]を、5.3 AMOTAと4.7 MOTAで上回った。また、現在の3Dトラッカーにおけるモーションモデルの品質を、2つの新しいメトリクスを評価することで研究する: 平均追跡速度誤差(ATVE)と追跡速度誤差(TVE)である。手作業で設計された関連付け手法と比較して、我々のエンドツーエンドの学習可能なトラッカーは、将来、自律走行分野において豊富なデータ量を享受できると信じている。


図3. FPSを1として、連続する8フレームでの可視化：鳥瞰図、フロントカメラ、フロント左カメラ、バック左カメラ。同じIDのオブジェクトは同じ色で塗られている。推定速度を矢印でプロットし、長い矢印は速度が大きいことを表す。私たちが示した例は、カメラをまたいで切り捨てられたオブジェクトを持つ複数のフレームを含んでいる。我々のアルゴリズムは、マルチカメラ特徴を自動的に融合し、切り捨てを正しく処理するように設計されている。

## References


[1] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3D Object Detection via Geometric Reasoning on Keypoint. *arXiv preprint arXiv:1905.05618*, 2019. 3

[2] Erkan Baser, Venkateshwaran Balasubramanian, Prarthana Bhattacharyya, and Krzysztof Czarnecki. FANTrack: 3D Multi-Object Tracking with Feature Association Network. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019. 2, 3

[3] Nuri Benbarka, Jona Schröder, and Andreas Zell. Score refinement for confidence-based 3D multi-object tracking. In *IROS*, 2021. 2

[4] Keni Bernardin, Alexander Elbs, and Rainer Stiefelhagen. Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment. In *ECCV Workshops*, 2006. 6

[5] Guillem Brasó and Laura Leal-Taixé. Learning a Neural Solver for Multiple Object Tracking. In *CVPR*, 2020. 2

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 5

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end Object Detection with Transformers. In *ECCV*, 2020. 3

[8] Mohamed Chaabane, Peter Zhang, J Ross Beveridge, and Stephen O'Hara. DEFT: Detection Embeddings for Tracking. *CVPR Workshops*, 2021. 1, 2, 3, 6

[9] Hsu-kuang Chiu, Jie Li, Rareş Ambruş, and Jeannette Bohg. Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving. In *ICRA*, 2021. 2

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. In *ICCV*, 2017. 6

[11] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular Quasi-Dense 3D Object Tracking. *arXiv preprint arXiv:2103.07351*, 2021. 1, 3, 6, 7

[12] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View. *arXiv preprint arXiv:2112.11790*, 2021. 3

[13] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv preprint arXiv:1509.04874*, 2015. 3

[14] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *CVPR*, 2019. 2

[15] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In *CVPR*, 2009. 6

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 6

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017. 3, 4

[18] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking Pseudo-LiDAR Representation. In *ECCV*, 2020. 3

[19] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape. In *CVPR*, 2019. 3

[20] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-Object Tracking with Transformers. *arXiv preprint arXiv:2101.02702*, 2021. 3

[21] Ziqi Pang, Zhichao Li, and Naiyan Wang. SimpleTrack: Understanding and Rethinking 3D Multi-object Tracking. *arXiv preprint arXiv:2111.09621*, 2021. 2, 5, 6, 7

[22] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is Pseudo-Lidar needed for Monocular 3D Object detection? In *ICCV*, 2021. 3

[23] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *ECCV*, 2020. 3

[24] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical Depth Distribution Network for Monocular 3D Object Detection. In *CVPR*, 2021. 3

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016. 3

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *NeurIPS*, 2015. 3

[27] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-Camera 3D Multi-Object Tracking Using Deep Learning Detections and PMBM Filtering. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 3

[28] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 2, 3

[29] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 3

[30] Xiao Tan, Zhigang Wang, Minyue Jiang, Xipeng Yang, Jian Wang, Yuan Gao, Xiangbo Su, Xiaoqing Ye, Yuchen Yuan, Dongliang He, et al. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In *CVPR Workshops*, 2019. 3

[31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*, 2019. 3

[32] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to Track with Object Permanence. In *ICCV*, 2021. 3

[33] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and Geometric Depth: Detecting Objects in Perspective. In *CoRL*, 2022. 3

[34] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection. *ICCV Workshops*, 2021. 3, 6

[35] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In *CVPR*, 2019. 3

[36] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries. In *CoRL*, 2021. 3, 4, 6, 7

[37] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-End Video Instance Segmentation with Transformers. In *CVPR*, 2021. 3

[38] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. *IROS*, 2020. 1, 2, 5, 6

[39] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning. In *CVPR*, 2020. 2

[40] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. In *ICIP*, 2017. 1

[41] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10), 2018. 2

[42] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3DSSD: Point-based 3D Single Stage Object Detector. In *CVPR*, 2020. 3

[43] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. *arXiv preprint arXiv:2006.11275*, 2020. 2, 5, 6, 7

[44] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. *arXiv preprint arXiv:1906.06310*, 2019. 3

[45] Jan-Nico Zaech, Alexander Liniger, Dengxin Dai, Martin Danelljan, and Luc Van Gool. Learnable Online Graph Representations for 3D Multi-Object Tracking. *IEEE Robotics and Automation Letters*, 2022. 2

[46] Fangao Zeng, Bin Dong, Tiancai Wang, Cheng Chen, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-End Multiple-Object Tracking with TRansformer. *arXiv preprint arXiv:2105.03247*, 2021. 3, 5

[47] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking Objects as Points. In *ECCV*, 2020. 3, 5, 6

[48] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. *arXiv preprint arXiv:1904.07850*, 2019. 3

[49] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*, 2018. 2