物体追跡導入によるリアルタイム・ゼロショットセグメンテーション

— 自律走行モビリティにおける認識・判断タスクへの応用 —

○馬場 琉生（千葉工大），上田 隆一（千葉工大），林原 靖男（千葉工大）

Real-time Zero-shot Segmentation through Object Tracking

— Application to Perception and Decision-making Tasks in Autonomous Mobility —

○ Ryusei BABA (CIT), Ryuichi UEDA (CIT) and Yasuo HAYASHIBARA (CIT)

Abstract: While zero-shot segmentation guided by natural language instructions offers high flexibility, it still faces challenges in ensuring real-time performance. In this study, we propose a method for real-time recognition by inserting a lightweight object tracking between object detection and segmentation. We implement this method on an autonomous vehicle and evaluate its applicability in real-world environments, focusing on vanishing point navigation and obstacle detection and stop tasks.

1. 緒言

近年，大規模基盤モデルの発展により，画像認識分野は大きな進歩を遂げている．例えば，テキストプロンプトから物体位置を特定する Grounding DINO [1] や，プロンプトに応じて物体を高精度にセグメンテーションする Segment Anything Model (SAM) [2] が挙げられる．この 2 つのモデルを組み合わせた Language Segment-Anything（Lang-SAM）[3] は，自然言語の指示のみで対象を切り出すゼロショットセグメンテーションを可能にする．

しかし，これらのモデルをリアルタイムシステムに応用しようとすると，処理を高頻度で繰り返す必要があるため，システム全体の処理速度が遅くなるという問題が生じる．この問題に対する有効なアプローチとして，高コストな検出と軽量な追跡を組み合わせる Tracking-by-Detection のパラダイムが知られている．このパラダイムは，毎フレーム実行される高精度な物体検出の結果を，軽量な追跡アルゴリズムでフレームをまたいで関連付けることにより，物体の ID を維持し，安定した追跡をリアルタイムで実現するものである．その代表的な手法に SORT [4] が存在する．

本研究では，この Tracking-by-Detection の考え方を応用し，言語指示によるゼロショットセグメンテーションをリアルタイム化するために，高コストな物体検出処理の頻度を削減し，その間を軽量な物体追跡で補間する手法を提案する．本手法を時速 6 [km/h] で走行する自律走行モビリティの消失点ナビゲーションおよび障害物検出・停止タスクへ応用し，実環境への適用可能性を調査する．

2. 提案手法

本研究では，**図 1** に示すように，高コストなゼロショットセグメンテーション処理と軽量な物体追跡処理を組み合わせる手法を提案する．これは Tracking-by-Detection の考え方を応用したものであるが，以下の点が異なる．

第一に，SORT などが毎フレームの物体検出を前提とするのに対し，本アプローチではゼロショットモデルから成る認識処理を意図的に間引いて実行する．これは，大規模基盤モデルの高い認識性能を維持しつつ，システム全体のスループットを確保するためである．

第二に，物体検出が行われないフレームでは，追跡処理が出力を補間する役割を担う．これにより，高周波な状態更新が求められるモビリティの制御タスクに対応できる可能性がある．

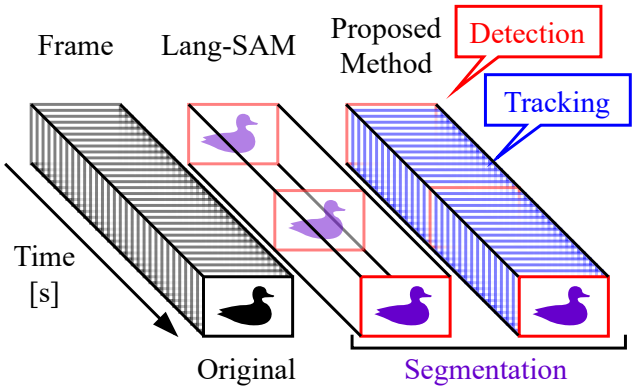

Fig. 1 Overview of the proposed method

3. システム構成

本章では，提案手法のシステム構成について述べる．**図 2** に示すように，本手法は，(1) 物体検出，(2) 物体追跡，(3) セグメンテーション，の 3 つのモジュールで構成される．以下にそれぞれのモジュールに関して述べる．

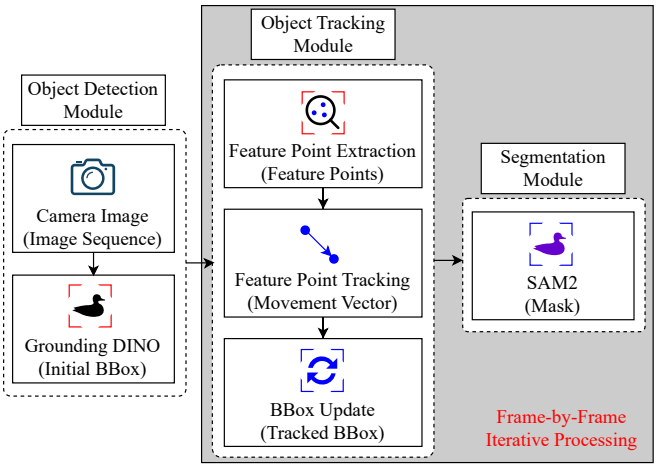

Fig. 2 Overview of the system

3.1 物体検出モジュール

テキストプロンプトに基づき，Grounding DINO を用いて物体検出を行う．これによって，テキストに対応するバウンディングボックス（以後，BBox と呼ぶ）が生成される．この処理は，毎フレームではなく一定間隔で実行し，計算負荷が過大とならないようにする．

3.2 物体追跡モジュール

後続のフレームでは，毎フレーム Grounding DINO を実行する代わりに，物体追跡で BBox を更新する．まず，前フレームの BBox 内で Shi-Tomasi 法 [5] に基づき画像特徴点を抽出する．次に，抽出した特徴点群を Lucas-Kanade 法 [6] に基づくピラミッドオプティカルフローで追跡し，その移動ベクトルの中央値を算出する．このベクトルを用いて BBox の位置と大きさを更新する．以上の一連の処理を繰り返し実行する．これらの手法は，標準的かつ計算効率に優れていることから，特徴点追跡に採用した．

3.3 セグメンテーションモジュール

検出または追跡によって更新された BBox に対し，SAM2 を用いてセグメンテーションマスクを生成する．これは，比較的小さいモデルを使用して毎フレーム実行する．これにより，検出から追跡，セグメンテーションまでの一連の処理を行う．

4. 提案手法の性能評価

4.1 性能評価の方法

本章では，提案手法の基本的な性能を明らかにするための評価実験について述べる．この実験の目的は，従来手法と比較して提案手法がどの程度処理速度を向上させるか，またその際にどの程度の検出位置の正確度を維持できるかを定量的に評価することである．実験環境の詳細を**表 1** に示す．コンピュータは後述のモビリティの制御に用いる NVIDIA Jetson AGX Orin (32GB) を使用し，MOT17 データセット [7] を使用した．また，テキストプロンプトには“pedestrian”を指定した．

Table 1 Experimental environment for evaluation

Computer	NVIDIA Jetson AGX Orin 32GB
Input Text	"pedestrian."
Grounding DINO	grounding-dino-base
SAM Model	sam2.1_hiera_small

4.2 結果と考察

評価指標には，システム全体の処理速度を示す FPS (Frames Per Second) と，BBox の正確度を示す IoU (Intersection over Union) の 2 つを用いた．評価データには，mot17-02-frcnn の 600 フレームを用いた．比較対象として，毎フレーム物体検出を行う従来手法（Lang-SAM）でも同様の評価を行った．

実験結果を**表 2** に示す．提案手法は従来手法に比べて処理速度を向上させる一方で，正確度の低下も確認された．この結果は，本手法において処理速度と正確度がトレードオフの関係にあることを示している．検出結果の一例を**図 3** に示す．図中では，正解データを赤色で，従来手法を青色で，提案手法を緑色で示している．図のように，完全一致ではないものの，提案手法により歩行者を概ね正しく検出できていることがわかる．

Table 2 Experimental results for evaluation

Method	FPS	IoU
Conventional	0.79	0.814
Proposed (Redetection interval: 2.0s)	2.15	0.707

Fig. 3 Comparison of Ground Truth, Lang-SAM, and the Proposed Method

5. AI-Formula への応用

実環境への適用可能性を調査するため，自律走行競技会 AI-Formula [8] における消失点ナビゲーションおよび障害物検出・停止タスクへ提案手法の応用を試みた．**図 4** に AI-Formula のモビリティを示す．モビリティには，ステレオカメラ，GNSS，IMU が搭載されている．本競技会では，実環境におけるリアルタイム認識と制御が要求される実践的な評価環境が提供されている．

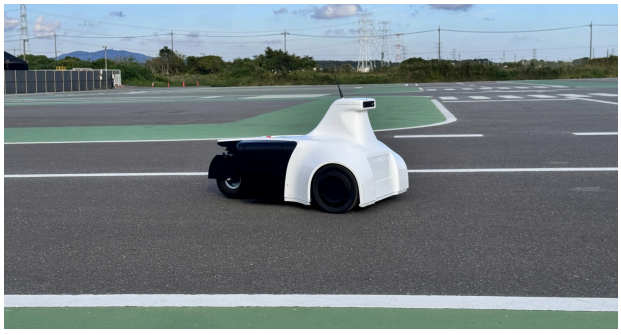

Fig. 4 AI-Formula mobility

5.1 実環境での実験方法

実環境への適用可能性を調査するため，モビリティがスタートラインから時速 6 [km/h] で経路を走行し，コースアウトせずに障害物の手前で停止できるかの挙動を確認する．実験環境の詳細は**表 3** の通りである．テキストプロンプトには，“white line. road. red pylon.” を指定して，その内の “white line” を消失点ナビゲーション，“red pylon” を障害物検出・停止タスクに用いた．走行経路は**図 5** に示す通り，破線の無い白線部分を対象とし，スタートから 20 [m] 先に障害物を設置した．

Table 3 An experimental environment for vanishing point navigation

Computer	NVIDIA Jetson AGX Orin 32GB
Middleware	ROS 2 Humble
Input Image	480x300
Input Text	"white line. road. red pylon."
Grounding DINO	grounding-dino-base
SAM Model	sam2.1_hiera_small

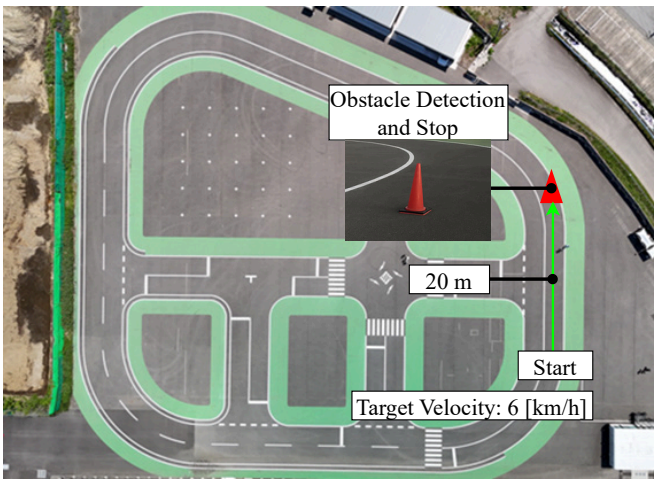

Fig. 5 Course and obstacle used in the experiment

提案手法によるセグメンテーションの結果を用いて，消失点ナビゲーションへ応用する実験を行う．ナビゲーションでは，最初に提案手法を用いてセグメンテーションされた白線を抽出する．次に，ハフ変換により 2 本の白線を求め，その消失点を算出する．最後に，算出した消失点が画像中心に位置するよう，PID 制御に基づきモビリティのヨー角速度を制御することで白線の消失点に向かって進むようにする．これによって，モビリティはレーンの白線に沿って経路を走行する．本手順による処理結果を**図 6** に示す．図中では，ハフ変換の出力を青線で，消失点を赤点で示している．

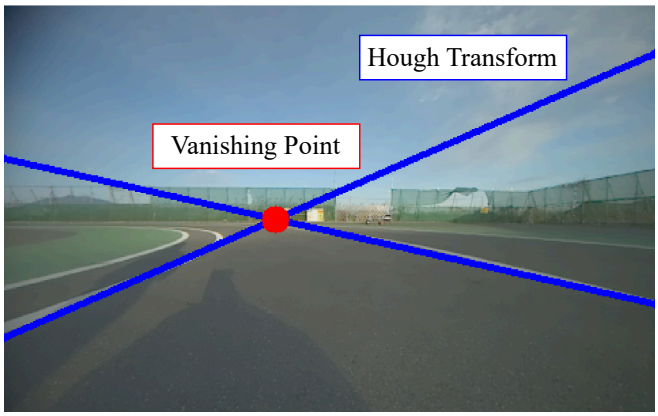

Fig. 6 An example of calculating the vanishing point

提案手法による物体検出および物体追跡の結果を用いて，障害物検出・停止タスクへ応用する実験を行う．実験では，**図 7** の赤いパイロンを障害物とする．モビリティがパイロンに接近し，検出した BBox の面積が事前に設定したしきい値を超えた場合に自動停止させる．しきい値は，実験開始前にモビリティの 5m 手前にパイロンを置き，そのときに検出される BBox の面積を計測して設定した．

Fig. 7 Obstacle detection using the BBox area as a threshold

5.2 実環境での実験結果

本実験では，時速 6 [km/h] で走行する自律走行モビリティを用いて，消失点ナビゲーションおよび障害物検出・停止タスクを実環境で行い，その適用可能性を調査した．その結果，モビリティは指定コースを逸脱することなく走行し，障害物手前で停止した．つまり，一例ではあるが，提案手法により事前学習なく環境を認識してコースを走行・停止できることを確認した．以下に，各タスクにおけるモビリティの挙動を詳述する．

図 8 に，GNSS データに基づき作成したモビリティの走行軌跡を示す．指定した約 20m のコースを逸脱することなく，一定速度で走行することを確認した．ただし，部分的に蛇行する挙動も見られた．これは，システムの処理負荷増大による制御周波数の低下が原因と考えられる．認識処理のみでは 2.15 FPS であった処理速度が，消失点ナビゲーションなどを含むシステム全体では低下したため，この処理遅延が蛇行につながったおそれがある．

図 9 に，障害物検出・停止タスクの結果を示す．事前に設定した目標停止距離 5 [m] に対し，モビリティは障害物の約 3.4 [m] 手前で停止し，目標と 1.6 [m] の誤差が生じた．これは，モビリティの制動距離に加え，システム全体の処理遅延も影響していると考えられる．

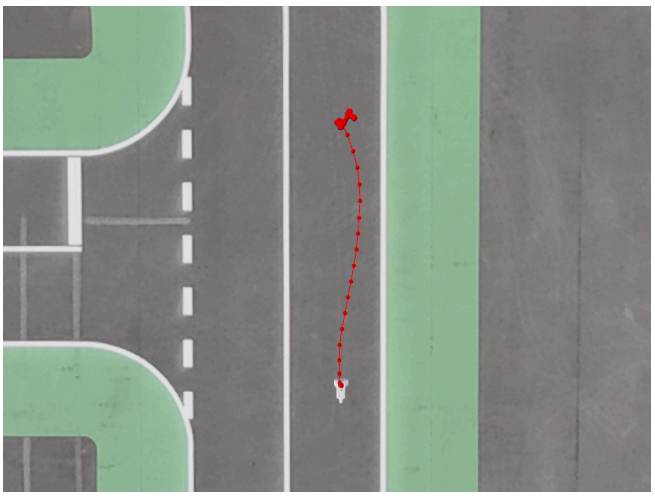

Fig. 8 The trajectory of mobility

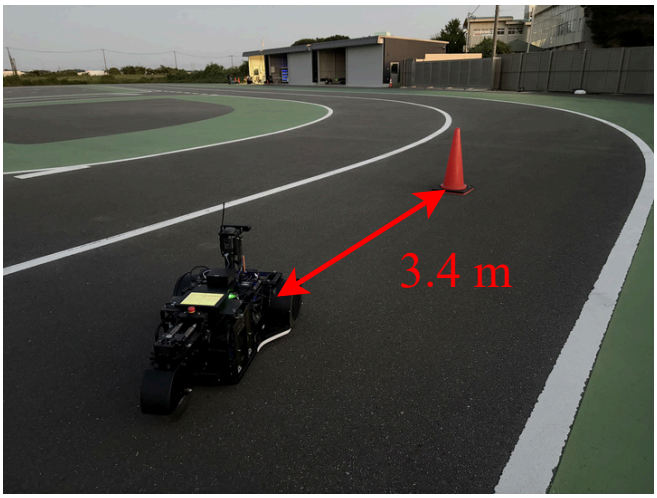

Fig. 9 Result of obstacle detection and stop

6. 結言

本研究では，自然言語の指示のみで画像から対象を切り出すゼロショットセグメンテーションに物体追跡を追加し，リアルタイム性を向上する手法を提案した．提案手法の評価では，処理速度を 2.7 倍に向上できた一方で，IoU が 0.107 低下するトレードオフも見られた．

また，実環境への適用可能性を調査するため，提案手法を AI-Formula に応用した．その結果，モビリティは指定したコースを逸脱せず，障害物の手前で停止した．今後は，さらなる高速化に取り組み，AI-Formula のレースや他のロボット技術チャレンジへの応用を目指す．

参考文献

[1] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Gao, C. Li, and L. Yuan. “Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection”. arXiv preprint arXiv:2303.05499, (2023).

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. “Segment Anything”. arXiv:2304.02643, (2023).

[3] L. Medeiros. “Lang-segment-anything”. GitHub repository. (2023). URL: https://github.com/luca-medeiros/lang-segment-anything.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. “Simple Online and Realtime Tracking”. *2016 IEEE International Conference on Image Processing (ICIP)*. (2016), pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.

[5] J. Shi and C. Tomasi. “Good features to track”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. (1994), pp. 593–600.

[6] B. D. Lucas and T. Kanade. “An iterative image registration technique with an application to stereo vision”. *Proceedings of the 7th international joint conference on Artificial intelligence*. Vol. 2. (1981), pp. 674–679.

[7] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. “MOT16: A Benchmark for Multi-Object Tracking”. arXiv preprint arXiv:1603.00831, (2016).

[8] M. Okada, I. Omura, Y. Akimoto, K. Sakazaki, S. Ebita, A. Kato, and Y. Yasui. “Operations and Platform Design for the AI-Formula”. *Proceedings of the 2025 SICE Festival with Annual Conference*. Honda R&D Co., Ltd. Chiang Mai, Thailand, (2025), pp. 1112–1115.