

車載カメラ動画像シミュレータにおける 対象物体マスク画像自動生成機能の実装

稲川 巧巳[†] 水野 倫太郎^{††}

[†] 明治大学総合数理学部ネットワークデザイン学科

中田 洋平^{†,††}

^{††} 明治大学大学院先端数理科学研究科

1. はじめに

現在、運転手のミスに起因する自動車交通事故を減らすべく、様々な取り組みが行われている。そのような中で、運転支援技術の発展に寄与することを目標として、著者らのグループでは、これまで車載カメラ動画像から運転手の注視点を予測するための動的顕著性マップの構築法の研究[1]や車載カメラ動画像を再現できるシミュレータの試作[2]を行ってきた。そして、同構築法で算出された動的顕著性マップを活用して、歩行者などの対象物体の誘目性を定量評価する手法も提案してきた[3]。ただし、同評価手法では、その算出過程で対象物体のマスク画像が必要となる。そのため、本稿では、前述のシミュレータにより再現された車載カメラ動画像を作成する際に、選択された対象物体のマスク画像を、より正確に生成できる機能の実装を行った。その内容について報告する。

2. 対象物体マスク画像自動生成機能の実装

本稿では、Unityで作成された車載カメラ動画像シミュレータ [2]において車載カメラを再現するために、Perceptionパッケージ[4]に搭載されているPerception Cameraを導入した。このPerception Cameraには、レンズに写る動画像だけでなく、物体検出やセマンティックセグメンテーションなどの学習タスクに必要な情報を出力する機能が存在する。図1.1は、このPerception Cameraにより出力された画像例を表している。また、対象物体を白色、それ以外を黒色となるように設定することで、対象物体の領域を示すマスク画像も生成することもできる。本稿では、このような方法によるマスク画像生成機能を実装した。

3. 動作検証

文献[3]で示されるように、同シミュレータを用いた場合には、対象物体のない背景画像を再現し、背景差分法によりマスク画像を生成することもできる。そこで、ベビーカーを押す人物モデルが歩行している場面でのマスク画像を、本稿で実装した機能と文献[3]に示す背景差分法で生成し、比較した。図2は、その比較結果である。一見すると、文献[3]に示される背景差分法によるもの(c)と、本稿

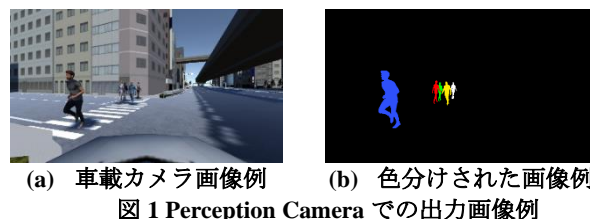


図1 Perception Cameraでの出力画像例

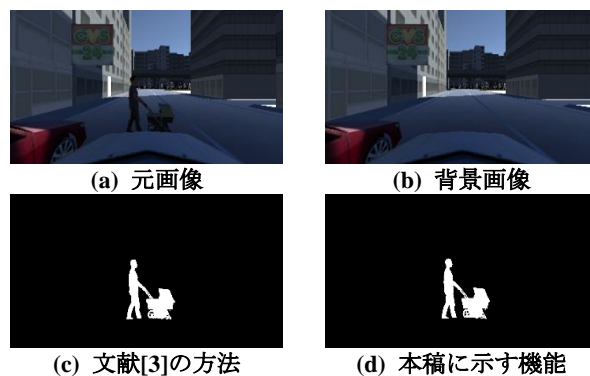


図2 マスク画像の比較

に示す機能によるもの(d)には大きな違いが見られない。ただし、ベビーカーの車輪のところなどに違いが見られ、文献[3]に示す方法によるものは、車輪の穴の部分が一部潰れていることが確認できる。また、対象物体に影がある場合には、文献[3]に示す方法では、影もマスク画像に含まれてしまうことがあった。このような動作検証により、本稿に示す機能では、文献[3]に示す方法よりも正確にマスク画像が生成できることを確認した。

参考文献

- [1] S. Nakazawa, S. Ushijima, and Y. Nakada, "An EM Algorithm based Method for Constructing Dynamic Saliency Maps Considering Characteristics while Driving", In Proc. of 10th Int. Symp. Inform. Commun. Technol., pp. 314-321, Dec. 2019.
- [2] R. Mizuno, S. Nakazawa, and Y. Nakada, "Simulator of Vehicle-Mounted Camera Video to Verify Dynamic Saliency Maps for Predicting Drivers' Gaze Points", Proc. of 2020 IEEE Symposium Series on Computational Intelligence, Paper ID 1139, pp. 1-6, Dec. 2021.
- [3] 稲川巧巳, 水野倫太郎, 中田洋平, "運転手注視点予測用動的顕著性マップを用いた対象物体に対する誘目性の定量評価法の検討", 第20回情報科学技術フォーラム (FIT2021), I-009, 2021年8月.
- [4] Unity Technologies, Perception Package (Unity Computer Vision): <https://github.com/Unity-Technologies/com.unity.perception>