

自動運転ソフトウェアを活用した周辺車両の半隠消表示

Half-DR Expression of Vehicle Periphery Utilizing Autonomous Driving Software

立命館大学 ○有富 友紀 竹村 岩朗, 池田 聖, 木村 朝子, 柴田 史久

Tomonori Aritomi, Iwao Takemura, Sei Ikeda, Asako Kimura, and Fumihisa Shibata
Ritsumeikan University

Abstract In this paper, on the premise that autonomous car spreads, we propose our system which make blind area semi-transparent (Half-DR) by utilizing autonomous driving software and sensors. On the condition of utilizing the function of existing modules such as SLAM of semi-transparent and visualize blind area of the transparent object which another vehicle detects. In addition, we conducted an experiment which aims to show the result of Half-DR and then explain the accuracy and limitations of our system.

1 はじめに

我々は自動運転車が普及するまでの過渡期もしくは普及後の社会を前提とし、搭載される共通のセンサ群を用い、隠消現実感 (Diminished Reality; DR) [1] 技術により死角を可視化 (半隠消表示) するシステムを試作した [2]. 本技術は、自動運転車の完全自動化前においては運転者への危険事象の判断材料の提供に、完全自動化においても、人工知能の危険回避動作の意図や原因を運転者に視覚的に伝えることや、仮想的な視界拡張により同乗者のストレスを軽減に利用できると考えられる. 本技術の原理は、図1に示すように他の車両 (前方車両) が死角の画像と3次元情報を観測し、利用者の車両 (自車両) の視界内に重畳することで、死角が可視化される.

上記DRシステムは、DRに都合の良い暗黙の仮定の入り込む余地のない状況で、自動走行用のソフトウェアおよびセンサのみで実現可能であることを確認するため、まず自動走行技術のテスト用に公開されているデータでシステムが正常に動作することを確認した[2]. しかし、画像取得位置と仮想視点が近いことや、透過領域の検出誤差が考慮されていないこと、公開データのキャリブレーション精度が悪いこと、生成画像の精度確認は不十分であった.

本稿では、より実際の状況に近い条件でDR画像を生成し、生成画像の精度等を確認する. 具体的には、センサのキャリブレーションを改善し、DRで想定される状況を模した車両配置を行い、死角を可視化する車両の検出を含めて可視化の性能を確認する.

2 自動走行ソフトウェアによるDR

既存システム[2]では、画像上での透過対象領域を与えると、死角の情報が重畳描画される. 車両に

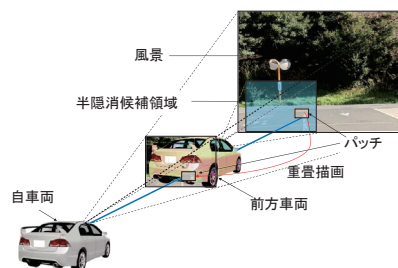


図 1: 複数車両を利用した死角の可視化

は距離センサLiDAR (Laser Imaging Detection and Ranging) とRGBカメラが固定されていることを前提とし、自動運転ソフトウェアには、国内の公道で自動運転の実績があるAutoware [3] が使用される. このソフトウェアには多数のモジュールが組み込まれており、そのうち、センサのデータを取得するモジュール、環境の密な点群生成と車両位置の推定にSLAMモジュール、点群を画像上に投影するProjectionモジュール等を主に用いる. これに、死角に対応する点群と画像パッチを他の視点の画像に重畳描画するDRモジュールを追加して死角の可視化を実現している.

これに加えて、本研究では複数視点の画像中での特徴点検出によりカメラの内部パラメータを推定可能なCOLMAP [6] を用い、LiDARに対する外部パラメータは、立方体の頂点をLiDARとカメラで観測することで対応付けPnP問題を解くことで求めた. これにより投影誤差6画素程度で内部パラメータの推定が可能である. Autowareのキャリブレーション機能では50画素以上であり、大幅に精度を改善できる.

また、今回の実験では、自動運転ソフトウェアに搭載されている車両および人物を検出する物体検出モジュールを活用し、半隠消候補領域を抽出す

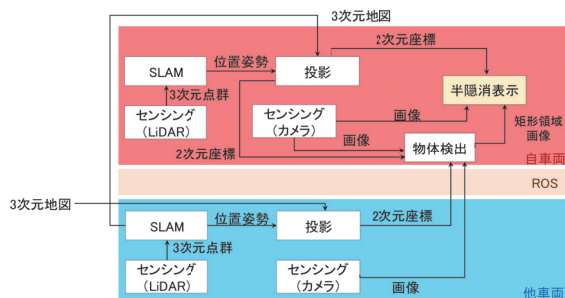


図 2: 既存モジュールとの依存関係

る。物体検出モジュールではSingle Shot Multibox Detector (SSD) [4] アルゴリズムが使用されており、予め学習した任意の物体の検出が可能である。今回は車両のみに対象を絞り、得られる矩形領域を拡大して死角の情報を重畳描画する領域とした。描画用のDRモジュールには領域の位置と大きさを送信することで、半隠消画像を自動的に生成する。物体認識による矩形領域の位置ずれを考慮して矩形領域は5画素拡大して用いた。

3 実験

図1に示すように屋外で車両を前後に配置し、自車両視点から前方車両を透過対象として検出し、背景像を自車両視点の画像に重畳した。LiDARとしてVelodyne社製VLP-16 (約300,000点 / 秒、5~20Hz, 水平360°, 垂直20°, 測定距離約100m), RGBカメラとしてPoint Grey Research社製Flea (解像度1280 × 1024, 60 fps, USB3.0) を搭載した。物体検出SDDは、深層学習ライブラリCaffe (Convolutional Architecture for Fast Feature Embedding) [5] 上で実装されたものを用いた。提案システムでは、図2のような車両間での直接的な通信を想定しているが、本実験では、自車両を死角が観測可能な位置に移動させ、仮想的な他車両として使用した。

図3(a)に車両の検出結果を示す。ちなみに、カメラ画像に3次元点群を投影誤差は、約10画素程度であった。カメラとLiDARのキャリブレーション誤差が約6画素であったため、約3画素程度の誤差がSLAMモジュールによる自車両の位置姿勢推定誤差が原因であると推測できる。人間が知覚できるレベルのズレではあるが、直接自車両から観測できる車両や歩行者を完全に消してしまう大きさの誤差ではないため、車両や歩行者の位置を示す用途には利用できると考えられる。

図3(b)は、上記投影点群を用いて前方車両による死角を半隠消候補領域として可視化した結果である。下部の観測されていない黒い部分の領域を除く



(a) 物体検出結果

(b) 可視化結果

図 3: 物体検出の様子

て、矩形領域の内部にアスファルト部分やオレンジ色のカーブミラーのポール、草木などの緑色の部分が描画できており、概ね背景領域の構造が把握できる精度で画像が生成されていることが分かる。しかし、草木とアスファルトの境などズレからも分かるように20画素未満の位置ずれが生じた。位置ずれの原因は、上記の点群投影に関する誤差に加えて点群をパッチで表現する際の描画精度によるものも含まれる。a)LiDARに加えて画像を使用した自己位置推定, b)GPS等を用いてSLAMし、3次元点群の精度向上, などにより改善の余地がある。

4 むすび

本論文では、既存の自動運転ソフトウェアのモジュールを活用し、周辺車両によって生じる死角を可視化する手法を提案し、実験により構造を概ね把握できる精度で死角の可視化ができることを確認した。今後、LiDARに加え画像やGPSを使用した自己位置推定、複数車両での死角観測, などにより、可視化画像の精度向上を検討する

参考文献

- [1] S. Mori *et al.*: A Survey of Diminished Reality: Techniques for Visually Concealing, Eliminating, and Seeing Through Real Objects, *IPJS Trans. on Computer Vision and Applications*, 9 (17), 2017.
- [2] 竹村 他: 自動運転ソフトウェアを活用した死角の半隠消表示, *VR学会大会講演論文集*, 14D-1, 2018.
- [3] S. Kato *et al.*: An Open Approach To Autonomous, *IEEE Micro* 35 (6) , 2015.
- [4] W. Lui *et al.*: SSD: Single Shot MultiBox Detector, *arXiv:1512.02325v5*, 2016.
- [5] Y. Jia *et al.*: Caffe: Convolutional Architecture for Fast Feature Embedding, *arXiv:1408.5093*, 2014.
- [6] COLMAP: <https://colmap.github.io/>
- [7] CloudCompare:<https://www.danielgm.net/cc/>